Panduan Pengguna

AWS OpsWorks



Versi API 2013-02-18

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

AWS OpsWorks: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Apa itu AWS OpsWorks? 1
AWS OpsWorks Jasa 1
AWS OpsWorks untuk Perusahaan Boneka 4
Region Support OpsWorks untuk Puppet Enterprise 5
Akhir Kehidupan FAQs 6
Bagaimana pelanggan yang sudah ada akan terpengaruh oleh End of Life ini? 6
Apa yang terjadi pada server saya jika saya tidak mengambil tindakan apa pun? 7
Apakah AWS OpsWorks for Puppet Enterprise menerima pelanggan baru? 7
Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama? 7
Tingkat dukungan teknis apa yang tersedia AWS OpsWorks for Puppet Enterprise? 7
Saya adalah pelanggan saat ini OpsWorks untuk Puppet Enterprise dan saya perlu meluncurkan server di akun yang tidak menggunakan layanan sebelumnya. Apakah saya bisa melakukan ini? 8
Apakah akan ada rilis fitur baru untuk AWS OpsWorks for Puppet Enterprise? 8
Memulai 8
Prasyarat 9
Buat Master Boneka 13
Selesai Konfigurasi 25
Tambahkan Node untuk Mengelola 29
Masuk ke Konsol Perusahaan Boneka 32
Opsional: Gunakan CodeCommit 36
Buat Master Boneka di CloudFormation 43
Prasyarat 44
Buat Master Perusahaan Boneka di AWS CloudFormation 44
Memperbarui Server untuk Menggunakan Domain Kustom 51
Prasyarat 52
Batasan 52
Memperbarui Server untuk Menggunakan Domain Kustom 53
Lihat Juga 56
Bekerja dengan Tag 57
Bagaimana Tag Bekerja di AWS OpsWorks for Puppet Enterprise 58
Menambahkan dan Mengelola Tag OpsWorks untuk Puppet Enterprise (Konsol) 59
Menambahkan dan Mengelola Tag OpsWorks untuk Puppet Enterprise (CLI) 62
Lihat Juga 67

Mencadangkan dan Memulihkan Server 67
Cadangkan OpsWorks untuk Server Perusahaan Boneka 68
Mengembalikan OpsWorks untuk Puppet Enterprise Server 72
Pemeliharaan Sistem 73
Mengkonfigurasi pemeliharaan sistem 74
Memulai pemeliharaan sistem sesuai permintaan 76
Memulihkan konfigurasi dan file khusus setelah pemeliharaan 77
Tambahkan Node Secara Otomatis 77
Langkah 1: Buat Peran IAM untuk Digunakan sebagai Profil Instance Anda 78
Langkah 2: Buat Instans dengan Menggunakan Skrip Asosiasi Tanpa Pengawasan 79
Hapus Node 80
Lihat Juga 81
Hapus seorang Puppet Master 81
Langkah 1: Pisahkan Node yang Dikelola 82
Langkah 2: Hapus Server 82
Lihat Juga 83
Migrasi Server Boneka ke Amazon EC2 83
Langkah 1: Hubungi Puppet untuk membeli lisensi 84
Langkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks 84
Langkah 3: Buat cadangan server Puppet Enterprise Anda OpsWorks 85
Langkah 4: Luncurkan EC2 instance baru 85
Langkah 5: Instal Puppet Enterprise pada instance baru EC2 87
Langkah 6: Kembalikan cadangan pada EC2 instance baru 87
Langkah 7: Konfigurasikan lisensi Boneka Anda 88
Langkah 8: Migrasikan node Anda 88
Langkah 9: Hapus server Puppet Enterprise Anda OpsWorks 90
Menggunakan AWS CloudTrail 91
OpsWorks untuk Informasi Perusahaan Boneka di CloudTrail 92
OpsWorks Memahami Entri Berkas Log Perusahaan Boneka 93
Pemecahan Masalah 95
Kiat pemecahan masalah umum 95
Memecahkan masalah kesalahan tertentu 96
Bantuan dan dukungan tambahan 101
AWS OpsWorks untuk Chef Automate 102
Region Support untuk AWS OpsWorks for Chef Automate 105
Akhir Kehidupan FAQs 106

Bagaimana pengguna yang ada akan terpengaruh oleh End of Life ini? 106
Apa yang terjadi pada server saya jika saya tidak mengambil tindakan apa pun? 107
Alternatif apa yang bisa saya transisi? 107
Apakah layanan ini masih menerima pelanggan baru? 107
Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama? 107
Tingkat dukungan teknis apa yang tersedia? 108
Saya adalah pelanggan saat ini OpsWorks untuk Chef Automate dan saya perlu meluncurkan server di akun yang sebelumnya tidak menggunakan layanan. Apakah saya bisa melakukan ini? 108
Apakah akan ada rilis fitur utama selama tahun depan? 108
Tingkatkan ke Chef Automate 2 108
Prasyarat untuk Upgrade ke Chef Automate 2 109
Tentang Proses Upgrade 109
Tingkatkan ke Chef Automate 2 (Konsol) 110
Tingkatkan ke Chef Automate 2 (CLI) 110
Kembalikan AWS OpsWorks for Chef Automate Server ke Chef Automate 1 (CLI) 111
Lihat Juga 112
Memulai 113
Prasyarat 113
Buat Server Otomatis Koki 116
Selesaikan konfigurasi dan unggah buku masak 129
Tambahkan node untuk mengelola 138
Masuk ke dasbor Chef Automate 145
Buat Server Otomatis Koki di CloudFormation 148
Prasyarat 149
Buat Server Otomatis Koki di AWS CloudFormation 150
Memperbarui Server untuk Menggunakan Domain Kustom 157
Prasyarat 158
Batasan 52
Memperbarui Server untuk Menggunakan Domain Kustom 53
Lihat Juga 56
Regenerasi starter kit 163
Regenerasi AWS OpsWorks for Chef Automate starter kit dengan AWS CLI 164
Bekerja dengan Tag 165
Bagaimana Tag Bekerja di AWS OpsWorks for Chef Automate 166
Tambahkan dan Kelola Tag di AWS OpsWorks for Chef Automate (Konsol) 167

Tambahkan dan Kelola Tag di AWS OpsWorks for Chef Automate (CLI) 170
Lihat Juga 175
Membuat cadangan dan memulihkan server 175
Cadangkan AWS OpsWorks for Chef Automate Server 176
Memulihkan AWS OpsWorks for Chef Automate Server 178
Pemeliharaan Sistem 180
Memastikan node mempercayai Otoritas AWS OpsWorks Sertifikasi 181
Mengkonfigurasi pemeliharaan sistem 182
Memulai pemeliharaan sistem sesuai permintaan 184
Memulihkan konfigurasi dan file khusus setelah pemeliharaan 184
Pemindaian Kepatuhan 185
Kepatuhan dalam Chef Automate 2.0 186
Kepatuhan dalam Chef Automate 1. x 194
Pembaruan Kepatuhan 200
Profil Kepatuhan Komunitas dan Kustom 200
Lihat Juga 200
Hapus node 201
Topik Terkait 202
Hapus server Chef Automate 202
Langkah 1: Pisahkan Node yang Dikelola 203
Langkah 2: Hapus Server 203
Setel ulang kredensi Chef 204
Menggunakan AWS CloudTrail 205
AWS OpsWorks for Chef Automate Informasi di CloudTrail 206
Memahami Entri File AWS OpsWorks for Chef Automate Log 207
Pemecahan Masalah 209
Kiat pemecahan masalah umum 209
Memecahkan masalah kesalahan tertentu 210
Bantuan dan dukungan tambahan 217
Keamanan dalam Manajemen AWS OpsWorks Konfigurasi (CM) 218
Perlindungan Data 219
Integrasi dengan AWS Secrets Manager 220
Enkripsi data 221
Enkripsi saat Data Tidak Berpindah 221
Enkripsi Saat Data Berpindah 221
Manajemen kunci 221

Identity and Access Management 222
Audiens 222
Mengautentikasi Menggunakan Identitas 223
Mengelola Akses Menggunakan Kebijakan 226
Bagaimana AWS OpsWorks CM Bekerja dengan IAM 229
Contoh Kebijakan Berbasis Identitas 234
Pemecahan Masalah 238
Kebijakan terkelola AWS 240
Cross-service membingungkan deputi pencegahan di AWS OpsWorks CM 249
Privasi Lalu Lintas Kerja Internet 252
Pembuatan Log dan Pemantauan 253
Validasi Kepatuhan 253
Ketahanan 254
Keamanan Infrastruktur 255
Analisis Konfigurasi dan Kelemahan 255
Praktik Terbaik Keamanan 256
AWS OpsWorks Tumpukan 258
Tumpukan 261
Lapisan 261
Resep dan LifeCycle Acara 261
Instans 262
Aplikasi 264
Menyesuaikan tumpukan Anda 264
Manajemen Sumber Daya 265
Keamanan dan Izin 265
Pemantauan dan Pencatatan 266
CLI, SDK, dan Template AWS CloudFormation 266
Akhir Kehidupan FAQs 267
Bagaimana pelanggan yang sudah ada akan terpengaruh oleh End of Life ini? 267
Apakah AWS OpsWorks Stacks menerima pelanggan baru? 268
Ke mana saya harus memigrasikan tumpukan yang ada? 268
Bagaimana saya bisa menyimpan EC2 instance Amazon saya yang ada setelah End of Life? 268
Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama? 269
Tingkat dukungan teknis apa yang tersedia AWS OpsWorks Stacks? 269
Apakah akan ada rilis fitur baru untuk AWS OpsWorks Stacks? 269

Memigrasi aplikasi Anda ke Manajer Aplikasi Systems Manager 269
Cara kerja skrip 270
Prasyarat 270
Batasan 271
Memulai 272
Pertanyaan yang Sering Diajukan 287
Pemecahan Masalah 299
Menggunakan alat AWS OpsWorks Stacks Detach in Place 300
Bagaimana prosesnya bekerja 301
Batasan 303
Memulai 304
Memulai 313
Support Wilayah 314
Memulai: Contoh 315
Memulai: Linux 336
Memulai: Windows 368
Memulai: Buku Masak 403
Praktik Terbaik 437
Penyimpanan Perangkat Root 437
Mengoptimalkan Jumlah Server 440
Mengelola Izin 443
Mengelola dan Menyebarkan Aplikasi dan Buku Masak 446
Ketergantungan Buku Masak Kemasan Secara Lokal 456
Tumpukan 460
Migrasikan tumpukan dari -Classic EC2 462
Membuat Tumpukan Baru 464
Menjalankan Stack di VPC 473
Perbarui Tumpukan 485
Klon Tumpukan 486
Perintah Tumpukan 488
Menggunakan JSON Kustom 491
Hapus Tumpukan 494
Lapisan 498
OpsWorks Dasar-dasar Lapisan 500
Lapisan Elastic Load Balancing 516
Lapisan Layanan Amazon RDS 521

Lapisan Klaster ECS ... 527
Lapisan Kustom ... 534
Instalasi Package Per-layer ... 535
Instans ... 536
Menggunakan AWS OpsWorks Instans Stacks ... 537
Menggunakan Sumber Daya Komputasi yang Dibuat Di Luar AWS OpsWorks Tumpukan ... 600
Mengedit Konfigurasi Instance ... 649
Menghapus Instans AWS OpsWorks Stacks ... 651
Login dengan SSH ... 653
Masuk dengan RDP ... 656
Aplikasi ... 661
Menambahkan Aplikasi ... 662
Menerapkan Aplikasi ... 669
Mengedit Aplikasi ... 673
Menghubungkan ke Database ... 675
Menggunakan Variabel Lingkungan ... 677
Melewati Data ke Aplikasi ... 678
Menggunakan Kunci SSH Repositori Git ... 682
Menggunakan Domain Kustom ... 683
Menggunakan SSL ... 686
Buku Masak dan Resep ... 694
Repositori ... 695
Versi Chef ... 699
Versi Ruby ... 718
Memasang Buku Masak Kustom ... 719
Memperbarui Buku Masak Kustom ... 723
Resep Eksekusi ... 725
Manajemen Sumber Daya ... 734
Mendaftarkan Sumber Daya dengan Stack ... 736
Melampirkan dan Memindahkan Sumber Daya ... 741
Melepaskan Sumber Daya ... 747
Sumber Daya Deregistering ... 749
Tanda ... 752
Mengatur Tag di Stack Level ... 753
Mengatur Tag di Tingkat Lapisan ... 755
Mengelola Tag dengan AWS CLI ... 757

Keterbatasan Tag 758
Pemantauan 759
Menggunakan Amazon CloudWatch 759
Menggunakan AWS CloudTrail 772
Menggunakan CloudWatch Log Amazon 775
Menggunakan Amazon CloudWatch Events 780
Keamanan dan Izin 781
Mengelola izin 783
Mengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda 808
Pencegahan Deputi Bingung 813
Menentukan Izin untuk Aplikasi yang Berjalan pada instance EC2 817
Mengelola Akses SSH 821
Mengelola Pembaruan Keamanan 829
Menggunakan Grup Keamanan 831
Koki 12 Linux 834
Gambaran Umum 835
Pindah ke Chef 12 836
Sistem Operasi yang Didukung 837
Tipe Instans Yang Didukung 837
Informasi Selengkapnya 837
Pindah ke Data Bags 838
Versi Chef Sebelumnya 840
Chef 11.10 dan Versi Sebelumnya untuk Linux 841
Menggunakan AWS OpsWorks Stacks dengan Layanan AWS Lainnya 1277
Menggunakan Back-end Data Store 1278
ElastiCache Redis 1287
Menggunakan Amazon S3 Bucket 1301
Menggunakan AWS CodePipeline dengan AWS OpsWorks Stacks 1315
Menggunakan AWS OpsWorks Stacks CLI 1379
Buat sebuah Instance 1381
Menerapkan Aplikasi 1384
Daftar Aplikasi 1386
Perintah List 1387
Daftar Deployment 1388
Daftar Alamat IP Elastis 1390
Daftar Contoh 1390

Daftar Tumpukan 1392
Daftar Layers 1394
Jalankan Resep 1398
Instal Dependensi 1399
Perbarui Konfigurasi Stack 1400
Panduan Debugging dan Pemecahan Masalah 1401
Resep Debugging 1402
Masalah Debugging dan Pemecahan Masalah Umum 1419
AWS OpsWorks Agen Tumpukan CLI 1430
agent_report 1432
get_json 1433
instance_report 1437
list_commands 1438
run_command 1439
show_log 1440
stack_state 1441
AWS OpsWorks Referensi Tas Data Stacks 1444
Tas Data Aplikasi (aws_opsworks_app) 1448
Tas Data Perintah (aws_opsworks_command) 1452
Tas Data Cluster Amazon ECS (aws_opsworks_ecs_cluster) 1454
Tas Data Elastic Load Balancing (aws_opsworks_elastic_load_balancer) 1455
Tas Data Instance (aws_opsworks_instance) 1456
Tas Data Lapisan (aws_opsworks_layer) 1462
Tas Data Amazon RDS (aws_opsworks_rds_db_instance) 1464
Tas Data Tumpukan (aws_opsworks_stack) 1465
Tas Data Pengguna (aws_opsworks_user) 1467
OpsWorks Perubahan Agen 1469
Agen Chef 12 Rilis 1469
Rilis Agen Chef 11.10 1473
Sumber daya 1479
Panduan Referensi, Alat, dan Sumber Daya Dukungan 1479
AWS Kit Pengembangan Perangkat Lunak 1480
Perangkat Lunak Sumber Terbuka 1481
AWS OpsWorks sejarah dokumen 1482
Pembaruan sebelumnya 1490
..... mcdxciv

Apa itu AWS OpsWorks?

> **Important**
>
> AWS OpsWorks Layanan telah mencapai akhir masa pakai dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks adalah layanan manajemen konfigurasi yang membantu Anda mengkonfigurasi dan mengoperasikan aplikasi di perusahaan cloud dengan menggunakan Puppet atau Chef. AWS OpsWorks Tumpukan dan AWS OpsWorks for Chef Automate memungkinkan Anda menggunakan buku masak Chef dan solusi untuk manajemen konfigurasi, sedangkan OpsWorks untuk Puppet Enterprise memungkinkan Anda mengonfigurasi server master Puppet Enterprise di. AWS Puppet menawarkan seperangkat alat untuk menegakkan keadaan infrastruktur yang diinginkan, dan mengotomatiskan tugas sesuai permintaan.

AWS OpsWorks Jasa

AWS OpsWorks untuk Perusahaan Boneka

OpsWorks untuk Puppet Enterprise memungkinkan Anda membuat AWS server master Boneka yang dikelola. Server master Puppet mengelola node di infrastruktur Anda, menyimpan fakta tentang node tersebut, dan berfungsi sebagai repositori pusat untuk modul Puppet Anda. Modul adalah unit kode Puppet yang dapat digunakan kembali dan dapat dibagikan yang berisi instruksi tentang bagaimana infrastruktur Anda harus dikonfigurasi. Anda dapat mengunduh modul komunitas dari Puppet Forge, atau menggunakan Kit Pengembangan Boneka untuk membuat modul kustom Anda sendiri, lalu mengelola penerapannya dengan Puppet Code Manager.

OpsWorks untuk Puppet Enterprise menyediakan Puppet master yang dikelola sepenuhnya, serangkaian alat otomatisasi yang memungkinkan Anda untuk memeriksa, mengirimkan, mengoperasikan, dan membuktikan aplikasi Anda di masa depan, dan akses ke antarmuka pengguna yang memungkinkan Anda melihat informasi tentang node dan aktivitas Puppet Anda. OpsWorks untuk Puppet Enterprise memungkinkan Anda menggunakan Puppet untuk

mengotomatiskan cara node dikonfigurasi, diterapkan, dan dikelola, baik itu EC2 instans Amazon atau perangkat lokal. Master OpsWorks for Puppet Enterprise menyediakan otomatisasi full-stack dengan menangani tugas-tugas seperti konfigurasi perangkat lunak dan sistem operasi, instalasi paket, pengaturan database, manajemen perubahan, penegakan kebijakan, pemantauan, dan jaminan kualitas.

Karena OpsWorks untuk Puppet Enterprise mengelola perangkat lunak Puppet Enterprise, server Anda dapat dicadangkan secara otomatis pada waktu yang Anda pilih, selalu menjalankan versi Puppet AWS terbaru yang kompatibel, dan selalu memiliki pembaruan keamanan terkini yang diterapkan. Anda dapat menggunakan grup EC2 Auto Scaling Amazon untuk mengaitkan EC2 node Amazon baru dengan server Anda secara otomatis.

AWS OpsWorks untuk Chef Automate

AWS OpsWorks for Chef Automate memungkinkan Anda membuat AWS server Chef yang dikelola yang menyertakan fitur premium Chef Automate, dan menggunakan Chef DK dan perkakas Chef lainnya untuk mengelolanya. Server Chef mengelola node di lingkungan Anda, menyimpan informasi tentang node tersebut, dan berfungsi sebagai repositori pusat untuk buku masak Chef Anda. Buku masak berisi resep yang dijalankan oleh agen klien Chef Infra (`chef-client`) pada setiap node yang Anda kelola dengan menggunakan Chef. Anda dapat menggunakan alat Chef seperti `knife`dan Test Kitchen untuk mengelola node dan buku masak di server Chef dalam AWS OpsWorks for Chef Automate layanan.

Chef Automate adalah paket perangkat lunak server yang disertakan yang menyediakan alur kerja otomatis untuk penerapan berkelanjutan dan pemeriksaan kepatuhan. AWS OpsWorks for Chef Automate menginstal dan mengelola Chef Automate, Chef Infra, dan Chef InSpec dengan menggunakan satu instans Amazon Elastic Compute Cloud. Dengan AWS OpsWorks for Chef Automate, Anda dapat menggunakan buku masak Chef yang ditulis komunitas atau kustom tanpa membuat perubahan spesifik. AWS OpsWorks

Karena AWS OpsWorks for Chef Automate mengelola komponen Chef Automate pada satu instance, server Anda dapat dicadangkan secara otomatis pada waktu yang Anda pilih, selalu menjalankan Chef versi minor terbaru, dan selalu menerapkan pembaruan keamanan terkini. Anda dapat menggunakan grup EC2 Auto Scaling Amazon untuk mengaitkan EC2 node Amazon baru dengan server Anda secara otomatis.

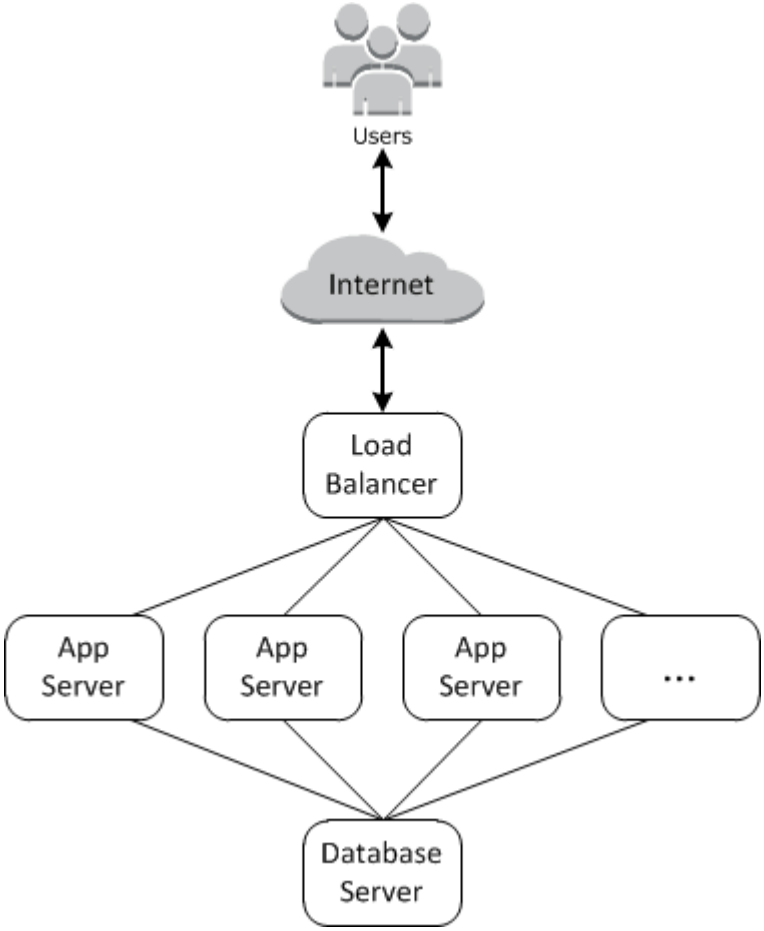
AWS OpsWorks Tumpukan

Komputasi berbasis cloud biasanya melibatkan grup sumber daya AWS, seperti EC2 instans dan instans Amazon Relational Database Service (RDS). Misalnya, aplikasi web biasanya

membutuhkan server aplikasi, server database, penyeimbang beban, dan sumber daya lainnya. Kelompok contoh ini biasanya disebut tumpukan.

AWS OpsWorks Stacks, layanan asli, menyediakan cara sederhana dan fleksibel untuk membuat dan mengelola tumpukan dan aplikasi. AWS OpsWorks Stacks memungkinkan Anda menyebarkan dan memantau aplikasi di tumpukan Anda. Anda dapat membuat tumpukan yang membantu Anda mengelola sumber daya cloud dalam grup khusus yang disebut lapisan. Lapisan mewakili satu set EC2 contoh yang melayani tujuan tertentu, seperti melayani aplikasi atau hosting server database. Lapisan bergantung pada resep Chef untuk menangani tugas seperti menginstal paket pada instance, menerapkan aplikasi, dan menjalankan skrip.

Tidak seperti AWS OpsWorks for Chef Automate, AWS OpsWorks Stacks tidak memerlukan atau membuat server Chef; AWS OpsWorks Stacks melakukan beberapa pekerjaan server Chef untuk Anda. AWS OpsWorks Stacks memantau kesehatan instans, dan menyediakan instans baru untuk Anda, bila perlu, dengan menggunakan Penyembuhan Otomatis dan Auto Scaling. Tumpukan server aplikasi sederhana mungkin terlihat seperti diagram berikut.



AWS OpsWorks untuk Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

OpsWorks untuk Puppet Enterprise memungkinkan Anda meluncurkan master Perusahaan Boneka dalam hitungan menit, dan memungkinkan AWS OpsWorks menangani operasi, pencadangan, pemulihan, dan peningkatan perangkat lunaknya. OpsWorks untuk Puppet Enterprise membebaskan Anda untuk fokus pada tugas manajemen konfigurasi inti, alih-alih mengelola master boneka. Dengan menggunakan OpsWorks untuk Puppet Enterprise, Anda dapat menggunakan konfigurasi yang sama untuk mengelola infrastruktur lokal dan cloud, membantu Anda menskalakan operasi secara efisien di lingkungan hybrid. Pengelolaan server master Puppet Anda disederhanakan oleh konsol Puppet Enterprise, the AWS Management Console, dan. AWS CLI

Master Puppet mengelola konfigurasi node di lingkungan Anda dengan menyajikan katalog konfigurasi untuk node tertentu ke `puppet-agent`perangkat lunak, dan berfungsi sebagai repositori pusat untuk modul Puppet Anda. Seorang Puppet master OpsWorks untuk Puppet Enterprise menyebarkan `puppet-agent` ke node terkelola Anda, dan menyediakan fitur premium dari Puppet Enterprise.

Master OpsWorks for Puppet Enterprise berjalan pada instans Amazon Elastic Compute Cloud. OpsWorks untuk server Puppet Enterprise dikonfigurasi untuk menjalankan versi terbaru Amazon Linux (Amazon Linux 2), dan versi terbaru dari Puppet Enterprise Master, versi 2019.8.5. Untuk informasi selengkapnya tentang perubahan dalam Puppet Enterprise 2019.8.5, lihat Catatan Rilis Perusahaan Boneka.

Ketika versi baru perangkat lunak Puppet tersedia, pemeliharaan sistem dirancang untuk memperbarui versi Puppet Enterprise di server secara otomatis, segera setelah melewati pengujian AWS. AWS melakukan pengujian ekstensif untuk memverifikasi bahwa upgrade Puppet siap produksi dan tidak mengganggu lingkungan pelanggan yang ada.

Anda dapat menghubungkan komputer atau EC2 instans lokal yang menjalankan sistem operasi yang didukung dan memiliki akses jaringan ke master OpsWorks for Puppet Enterprise. Perangkat lunak `puppet`agen diinstal oleh master Puppet pada node yang ingin Anda kelola.

Topik

- Region Support OpsWorks untuk Puppet Enterprise
- AWS OpsWorks for Puppet Enterprise Akhir Kehidupan FAQs
- Memulai dengan OpsWorks untuk Perusahaan Boneka
- Buat AWS OpsWorks for Puppet Enterprise Master dengan menggunakan AWS CloudFormation
- Memperbarui Server Perusahaan Boneka OpsWorks untuk Menggunakan Domain Kustom
- Bekerja dengan Tag pada AWS OpsWorks for Puppet Enterprise Sumber Daya
- Cadangkan dan Kembalikan Server OpsWorks untuk Puppet Enterprise
- Pemeliharaan Sistem OpsWorks untuk Perusahaan Boneka
- Menambahkan Node Secara Otomatis OpsWorks untuk Perusahaan Boneka
- Putuskan Node dari Server OpsWorks for Puppet Enterprise
- Hapus OpsWorks untuk Puppet Enterprise Server
- Cara memigrasikan server OpsWorks for Puppet Enterprise ke Amazon Elastic Compute Cloud (Amazon) EC2
- Logging OpsWorks untuk Panggilan API Perusahaan Boneka dengan AWS CloudTrail
- Pemecahan Masalah OpsWorks untuk Perusahaan Boneka

Region Support OpsWorks untuk Puppet Enterprise

Titik akhir regional berikut mendukung OpsWorks para master Puppet Enterprise. OpsWorks untuk Puppet Enterprise menciptakan sumber daya yang terkait dengan master Wayang Anda, seperti profil instans, pengguna, dan peran layanan, di titik akhir regional yang sama dengan master Wayang Anda. Master boneka Anda harus berada di VPC. Anda dapat menggunakan VPC yang Anda buat atau sudah miliki, atau menggunakan VPC default.

- Wilayah AS Timur (Ohio)
- Wilayah AS Timur (Virginia Utara)
- Wilayah AS Barat (California Utara)
- Wilayah US West (Oregon)

- Wilayah Asia Pasifik (Tokyo)
- Wilayah Asia Pasifik (Singapore)
- Wilayah Asia Pasifik (Sydney)
- Wilayah Eropa (Frankfurt)
- Wilayah Eropa (Irlandia)

AWS OpsWorks for Puppet Enterprise Akhir Kehidupan FAQs

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik

- Bagaimana pelanggan yang sudah ada akan terpengaruh oleh End of Life ini?
- Apa yang terjadi pada server saya jika saya tidak mengambil tindakan apa pun?
- Apakah AWS OpsWorks for Puppet Enterprise menerima pelanggan baru?
- Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama?
- Tingkat dukungan teknis apa yang tersedia AWS OpsWorks for Puppet Enterprise?
- Saya adalah pelanggan saat ini OpsWorks untuk Puppet Enterprise dan saya perlu meluncurkan server di akun yang tidak menggunakan layanan sebelumnya. Apakah saya bisa melakukan ini?
- Apakah akan ada rilis fitur baru untuk AWS OpsWorks for Puppet Enterprise?

Bagaimana pelanggan yang sudah ada akan terpengaruh oleh End of Life ini?

Pelanggan yang ada tidak akan terpengaruh hingga 31 Maret 2024, tanggal Akhir Kehidupan OpsWorks untuk Perusahaan Boneka. Setelah tanggal End of Life, pelanggan tidak akan lagi dapat mengelola server mereka menggunakan OpsWorks konsol atau API.

Apa yang terjadi pada server saya jika saya tidak mengambil tindakan apa pun?

Mulai 31 Maret 2024, Anda tidak akan lagi dapat mengelola server Anda menggunakan OpsWorks konsol atau API. Pada saat itu, kami akan berhenti melakukan fungsi manajemen yang sedang berlangsung untuk server Anda seperti pencadangan atau pemeliharaan. Untuk membatasi dampak bagi pelanggan, kami akan membiarkan EC2 instans berjalan yang mencadangkan server Puppet Enterprise, tetapi lisensi mereka tidak lagi berlaku karena penggunaan tidak lagi tercakup (atau ditagih) berdasarkan perjanjian layanan OpsWorks for Puppet Enterprise. Jika Anda ingin terus mengelola infrastruktur dengan Puppet Enterprise, lihat Cara memigrasikan server OpsWorks for Puppet Enterprise ke Amazon Elastic Compute Cloud (Amazon). EC2

Apakah AWS OpsWorks for Puppet Enterprise menerima pelanggan baru?

Tidak. AWS OpsWorks for Puppet Enterprise tidak lagi menerima pelanggan baru.

Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama?

Ya. API dan Konsol akan mencapai Akhir Kehidupan dan tidak dapat digunakan mulai 31 Maret 2024 di semua Wilayah. Untuk daftar Wilayah AWS tempat AWS OpsWorks for Puppet Enterprise tersedia, lihat Daftar Layanan AWS Regional.

Tingkat dukungan teknis apa yang tersedia AWS OpsWorks for Puppet Enterprise?

AWS akan terus memberikan tingkat dukungan yang sama untuk AWS OpsWorks for Puppet Enterprise yang dimiliki pelanggan hari ini hingga tanggal Akhir Kehidupan. Jika Anda memiliki pertanyaan atau masalah, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Saya adalah pelanggan saat ini OpsWorks untuk Puppet Enterprise dan saya perlu meluncurkan server di akun yang tidak menggunakan layanan sebelumnya. Apakah saya bisa melakukan ini?

Umumnya tidak, kecuali ada keadaan luar biasa untuk melakukannya. Jika Anda memiliki situasi khusus, hubungi AWS Dukungan Tim di AWS Re:Post atau melalui AWS Dukungan Premium dengan rincian dan pembenaran untuk ini dan kami akan meninjau permintaan Anda.

Apakah akan ada rilis fitur baru untuk AWS OpsWorks for Puppet Enterprise?

Tidak. Karena layanan ini mencapai End of Life, kami tidak akan merilis fitur baru apa pun. Namun, kami akan terus melakukan peningkatan keamanan dan mengelola server seperti yang diharapkan hingga tanggal End of Life.

Memulai dengan OpsWorks untuk Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

OpsWorks untuk Puppet Enterprise memungkinkan Anda menjalankan server Puppet Enterprise di. AWS Anda dapat menyediakan server master Puppet Enterprise dalam waktu sekitar 15 menit.

Mulai 3 Mei 2021, OpsWorks untuk Puppet Enterprise menyimpan beberapa atribut server Puppet Enterprise di. AWS Secrets Manager Untuk informasi selengkapnya, lihat Integrasi dengan AWS Secrets Manager.

Panduan berikut membantu Anda membuat master Boneka pertama Anda OpsWorks untuk Puppet Enterprise.

Prasyarat

Sebelum Anda mulai, Anda harus menyelesaikan prasyarat berikut.

Topik

- Pasang Kit Pengembangan Boneka
- Instal Alat Klien Perusahaan Boneka
- Mengatur Repositori Kontrol Git
- Siapkan VPC
- Mengatur Pasangan EC2 Kunci (Opsional)
- Prasyarat untuk Menggunakan Domain Kustom (Opsional)

Pasang Kit Pengembangan Boneka

1. Dari situs web Puppet, unduh Kit Pengembangan Boneka yang cocok dengan sistem operasi komputer lokal Anda.
2. Instal Kit Pengembangan Boneka.
3. Tambahkan Kit Pengembangan Boneka ke `PATH` variabel komputer lokal Anda.

   - Pada sistem operasi Linux atau macOS, Anda dapat menambahkan Kit Pengembangan Boneka ke `PATH` variabel Anda dengan menjalankan perintah berikut di shell Bash.

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

   - Pada sistem operasi berbasis Windows, Anda dapat menambahkan Kit Pengembangan Boneka ke **`PATH`** variabel Anda dengan menggunakan perintah.NET Framework berikut dalam PowerShell sesi, atau di kotak dialog Variabel Lingkungan yang dapat diakses dari System Properties. Anda mungkin perlu menjalankan PowerShell sesi Anda sebagai administrator untuk menjalankan perintah berikut.

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

Instal Alat Klien Perusahaan Boneka

Alat klien Puppet Enterprise (PE) adalah seperangkat alat baris perintah yang memungkinkan Anda mengakses layanan Puppet Enterprise dari workstation Anda. Alat dapat diinstal pada banyak sistem operasi yang berbeda, dan mereka juga dapat diinstal pada node yang Anda kelola dengan menggunakan Puppet. Untuk informasi tentang sistem operasi yang didukung untuk alat, dan cara menginstalnya, lihat Menginstal alat klien PE di dokumentasi Puppet Enterprise.

Mengatur Repositori Kontrol Git

Sebelum Anda dapat meluncurkan master Puppet, Anda harus memiliki repositori kontrol yang dikonfigurasi di Git untuk menyimpan dan mengubah modul dan kelas Puppet Anda. URL ke repositori Git dan informasi akun HTTPS atau SSH untuk mengakses repositori diperlukan dalam langkah-langkah untuk meluncurkan server master Puppet Enterprise Anda. Untuk informasi selengkapnya tentang cara mengatur repositori kontrol yang akan digunakan master Perusahaan Boneka Anda, lihat Menyiapkan repositori kontrol. Anda juga dapat menemukan instruksi pengaturan repositori kontrol di readme untuk repositori sampel Puppet di. `control-repo` GitHub Struktur repositori kontrol menyerupai yang berikut ini.

```
### 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
```

Menyiapkan repositori dengan menggunakan CodeCommit

Anda dapat membuat repositori baru dengan menggunakan. CodeCommit Untuk informasi selengkapnya tentang cara menggunakan CodeCommit untuk membuat repositori kontrol Anda, lihat the section called “Opsional: Gunakan CodeCommit” di panduan ini. Untuk informasi selengkapnya tentang cara memulai Git aktif CodeCommit, lihat Memulai AWS CodeCommit. Untuk mengotorisasi server Puppet Enterprise untuk repositori Anda, lampirkan `AWSCodeCommitReadOnly` kebijakan tersebut ke peran profil instans IAM Anda. OpsWorks

Siapkan VPC

Master OpsWorks for Puppet Enterprise Anda harus beroperasi di Amazon Virtual Private Cloud. Anda dapat menambahkannya ke VPC yang ada, menggunakan VPC default, atau membuat VPC baru untuk memuat server. Untuk informasi tentang Amazon VPC dan cara membuat VPC baru, lihat Panduan Memulai VPC Amazon.

Jika Anda membuat VPC Anda sendiri, atau menggunakan yang sudah ada, VPC harus memiliki pengaturan atau properti berikut.

- VPC harus memiliki setidaknya satu subnet.

  Jika master OpsWorks for Puppet Enterprise Anda dapat diakses publik, buat subnet publik, dan aktifkan Auto-assign IP publik.
- Resolusi DNS harus diaktifkan.
- Pada subnet, aktifkan Auto-assign IP publik.

Mengatur Pasangan EC2 Kunci (Opsional)

Koneksi SSH tidak diperlukan atau direkomendasikan untuk manajemen tipikal server Puppet; Anda dapat menggunakan AWS Management Console dan AWS CLI perintah untuk melakukan banyak tugas manajemen di server Puppet Anda.

Sebuah EC2 key pair diperlukan untuk terhubung ke server Anda dengan menggunakan SSH jika Anda kehilangan atau ingin mengubah password login untuk konsol berbasis web Puppet Enterprise. Anda dapat menggunakan key pair yang sudah ada, atau membuat key pair baru. Untuk informasi selengkapnya tentang cara membuat EC2 key pair baru, lihat Amazon EC2 Key Pairs.

Jika Anda tidak membutuhkan EC2 key pair, Anda siap untuk membuat master Puppet Enterprise.

Prasyarat untuk Menggunakan Domain Kustom (Opsional)

Anda dapat mengatur master Puppet Enterprise di domain Anda sendiri, menentukan titik akhir publik dalam domain khusus untuk digunakan sebagai titik akhir server Anda. Saat Anda menggunakan domain khusus, semua hal berikut diperlukan, seperti yang dijelaskan secara rinci di bagian ini.

Topik

- Mengatur Domain Kustom
- Dapatkan Sertifikat
- Dapatkan Kunci Pribadi

Mengatur Domain Kustom

Untuk menjalankan master Puppet Enterprise Anda di domain kustom Anda sendiri, Anda akan memerlukan titik akhir publik dari server, seperti. `https://aws.my-company.com` Jika Anda menentukan domain kustom, Anda juga harus memberikan sertifikat dan kunci pribadi, seperti yang dijelaskan di bagian sebelumnya.

Untuk mengakses server setelah Anda membuatnya, tambahkan catatan DNS CNAME di layanan DNS pilihan Anda. Catatan ini harus mengarahkan domain kustom ke titik akhir (nilai `Endpoint` atribut server) yang dihasilkan oleh proses pembuatan master Puppet. Anda tidak dapat mengakses server menggunakan nilai `Endpoint` jika server menggunakan domain kustom.

Dapatkan Sertifikat

Untuk mengatur master boneka Anda di domain kustom Anda sendiri, Anda memerlukan sertifikat HTTPS berformat PEM. Ini bisa berupa sertifikat tunggal yang ditandatangani sendiri, atau rantai sertifikat. Saat Anda menyelesaikan alur kerja Create a Puppet Enterprise Master, jika Anda menentukan sertifikat ini, Anda juga harus menyediakan domain kustom dan kunci pribadi.

Berikut ini adalah persyaratan untuk nilai sertifikat:

- Anda dapat memberikan sertifikat yang ditandatangani sendiri, kustom, atau rantai sertifikat penuh.
- Sertifikat harus merupakan sertifikat X509 yang valid, atau rantai sertifikat dalam format PEM.
- Sertifikat harus valid pada saat diunggah. Sertifikat tidak dapat digunakan sebelum periode validitasnya dimulai (tanggal `NotBefore` sertifikat) atau setelah kedaluwarsa (tanggal `NotAfter` sertifikat).

- Nama umum sertifikat atau nama alternatif subjek (SANs), jika ada, harus sesuai dengan nilai domain kustom.
- Sertifikat harus sesuai dengan nilai bidang kunci pribadi kustom.

Dapatkan Kunci Pribadi

Untuk mengatur master boneka Anda di domain kustom Anda sendiri, Anda memerlukan kunci pribadi dalam format PEM untuk menghubungkan ke server dengan menggunakan HTTPS. Kunci privat tidak boleh dienkripsi; tidak dapat dilindungi oleh kata sandi atau frasa sandi. Jika Anda menentukan kunci pribadi khusus, Anda juga harus menyediakan domain khusus dan sertifikat.

Buat Master Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

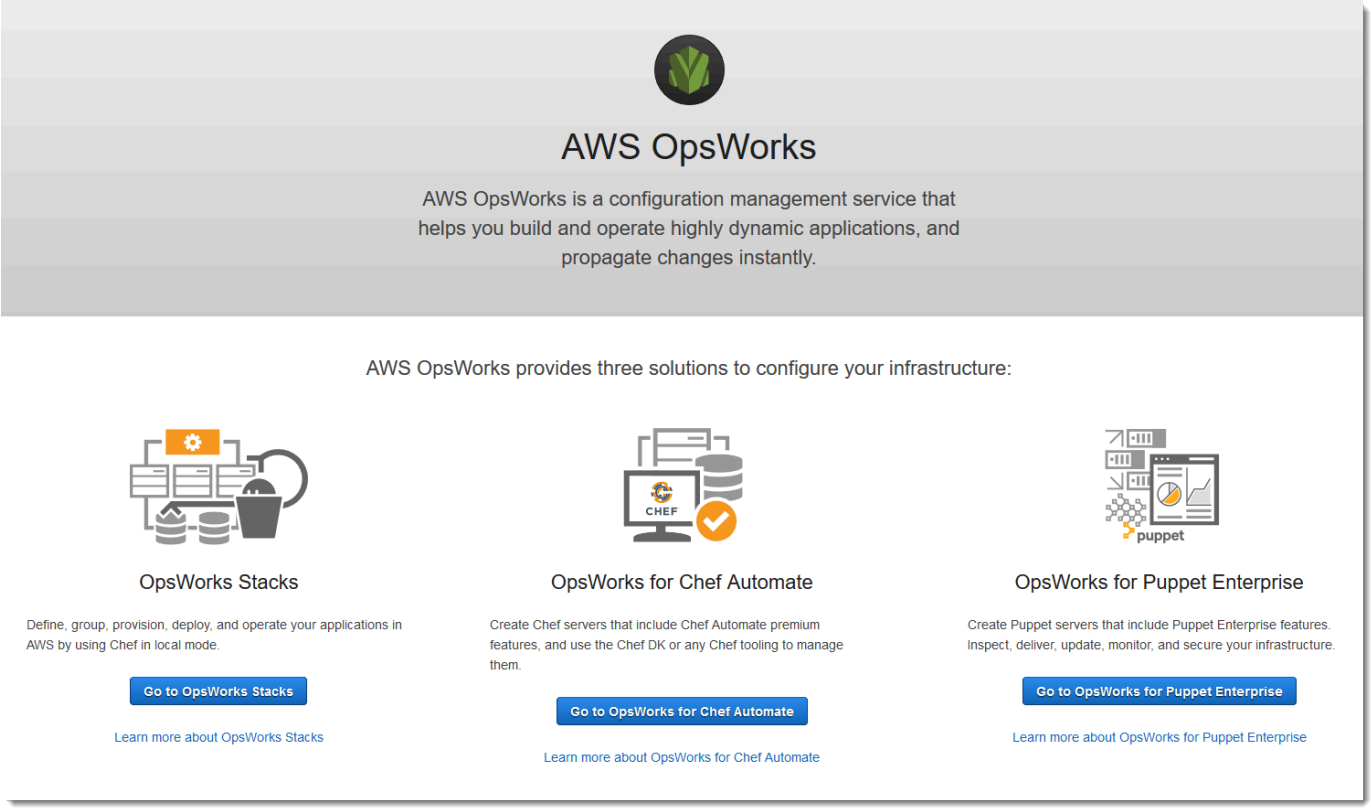
Anda dapat membuat master Puppet dengan menggunakan konsol OpsWorks for Puppet Enterprise, atau. AWS CLI

Topik

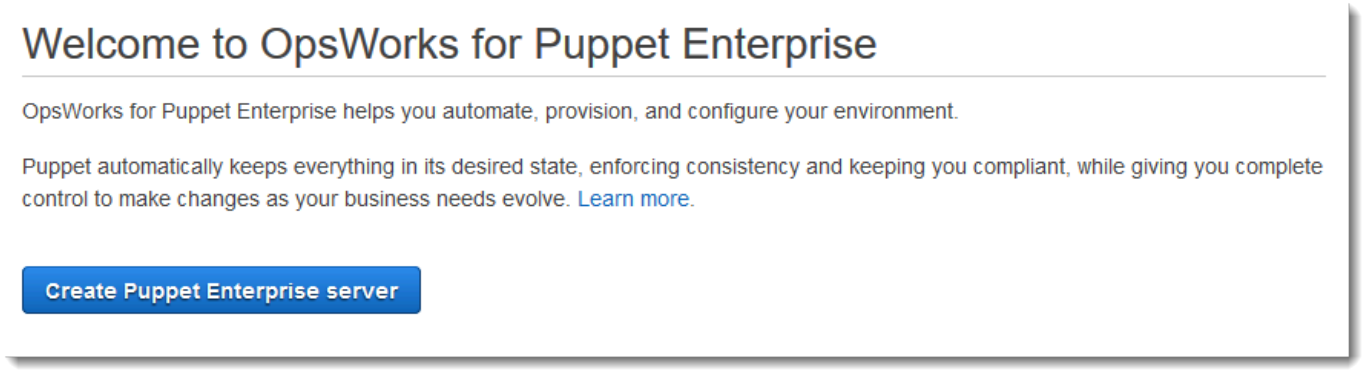
- Buat Master Perusahaan Boneka dengan menggunakan AWS Management Console
- Buat Master Perusahaan Boneka dengan menggunakan AWS CLI

Buat Master Perusahaan Boneka dengan menggunakan AWS Management Console

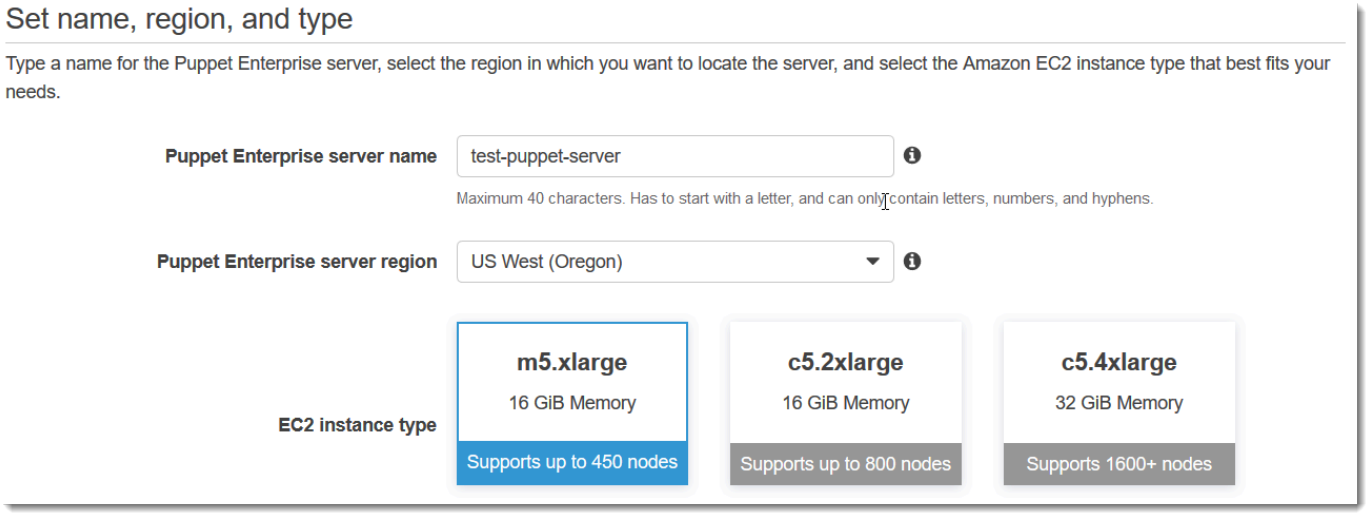
1. Masuk ke AWS Management Console dan buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di AWS OpsWorks halaman beranda, pilih Go to OpsWorks for Puppet Enterprise.



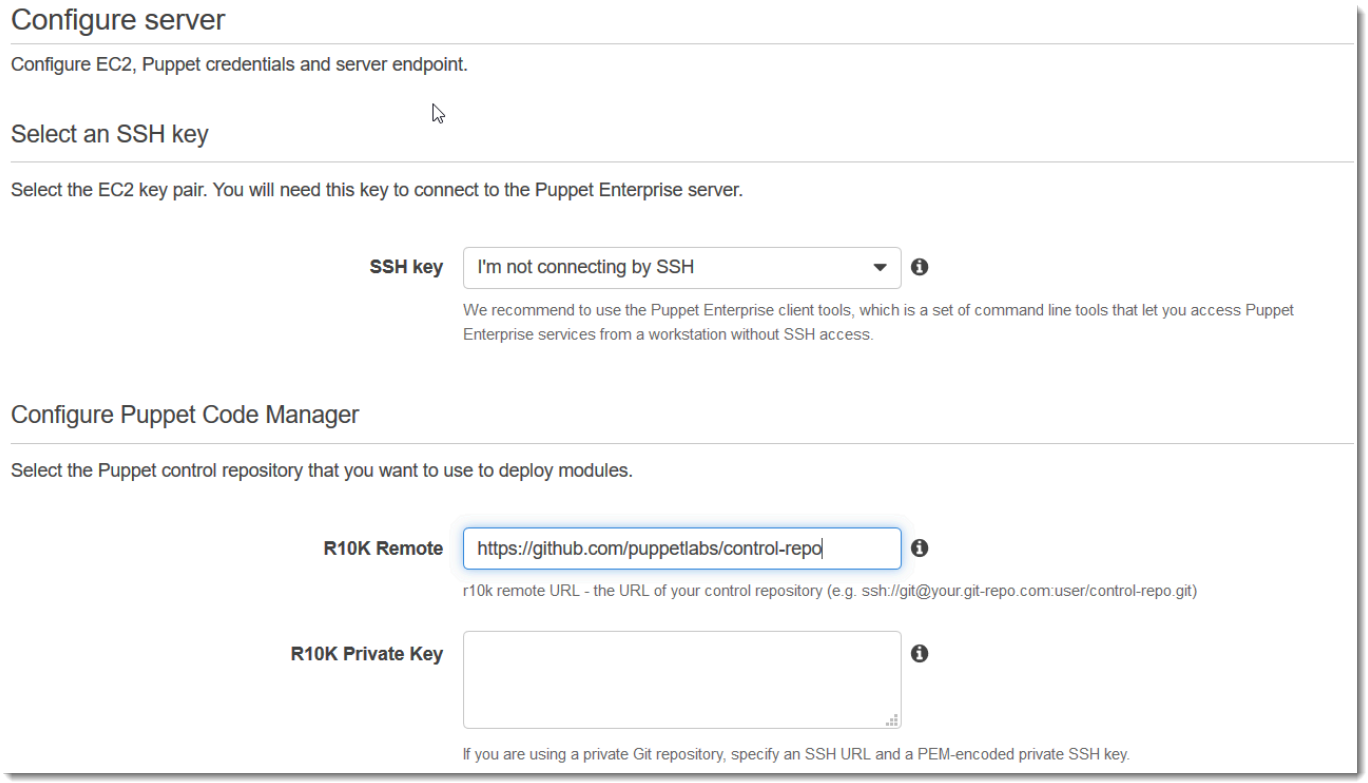
3. Pada halaman beranda OpsWorks for Puppet Enterprise, pilih server Create Puppet Enterprise.



4. Pada halaman Set name, region, dan type, tentukan nama untuk server Anda. Nama master boneka bisa maksimal 40 karakter, harus dimulai dengan huruf, dan hanya dapat berisi karakter alfanumerik dan tanda hubung. Pilih wilayah yang didukung, lalu pilih jenis instance yang mendukung jumlah node yang ingin Anda kelola. Anda dapat mengubah jenis instance setelah server Anda dibuat, jika diperlukan. Untuk panduan ini, kami membuat jenis instance m5.xlarge di Wilayah AS Barat (Oregon). Pilih Berikutnya.



5. Pada halaman Configure server, tinggalkan pilihan default di daftar drop-down kunci SSH, kecuali jika Anda ingin menentukan nama key pair. Di bidang jarak jauh r10k dari area Configure Puppet Code Manager, tentukan URL SSH atau HTTPS yang valid dari remote Git Anda. Di bidang kunci pribadi r10k, tempel di kunci pribadi SSH yang AWS OpsWorks dapat digunakan untuk mengakses repositori jarak jauh r10k. Ini disediakan oleh Git ketika Anda membuat repositori pribadi, tetapi tidak diperlukan jika Anda menggunakan otentikasi HTTPS untuk mengakses repositori kontrol Anda. Pilih Berikutnya.

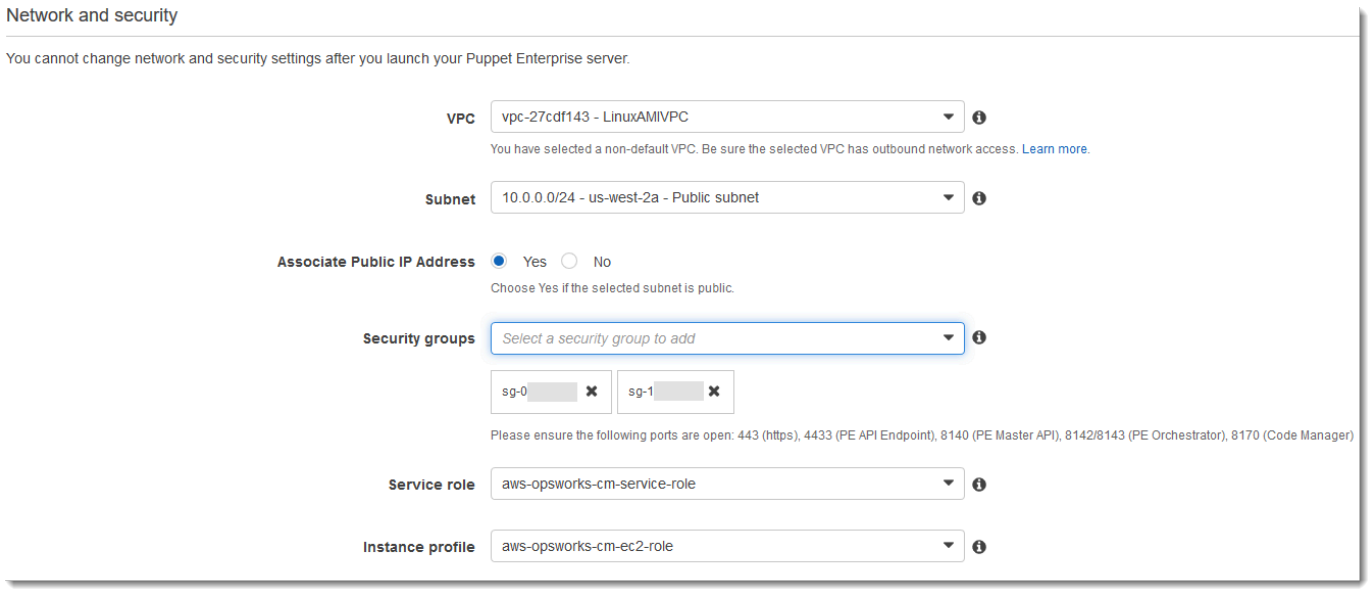


6. Untuk Tentukan titik akhir server, biarkan default, Gunakan titik akhir yang dihasilkan secara otomatis lalu pilih Berikutnya, kecuali jika Anda ingin server Anda berada di domain kustom Anda sendiri. Untuk mengonfigurasi domain khusus, lanjutkan ke langkah berikutnya.
7. Untuk menggunakan domain kustom, untuk Menentukan titik akhir server, pilih Gunakan domain kustom dari daftar drop-down.

   a. Untuk nama domain yang sepenuhnya memenuhi syarat (FQDN), tentukan FQDN. Anda harus memiliki nama domain yang ingin Anda gunakan.

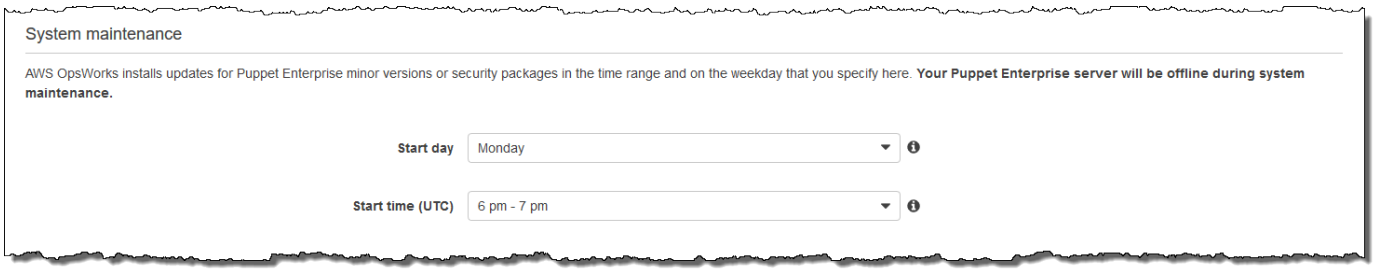
   b. Untuk sertifikat SSL, tempel di seluruh sertifikat berformat PEM, dimulai dengan dan diakhiri dengan`-----BEGIN CERTIFICATE-----`. `-----END CERTIFICATE-----` Subjek sertifikat SSL harus sesuai dengan FQDN yang Anda masukkan pada langkah sebelumnya. Hapus baris tambahan sebelum dan sesudah sertifikat.

   c. Untuk kunci pribadi SSL, tempel di seluruh kunci pribadi RSA, dimulai dengan `-----BEGIN RSA PRIVATE KEY-----` dan diakhiri dengan. `-----END RSA PRIVATE KEY-----` Kunci pribadi SSL harus sesuai dengan kunci publik dalam sertifikat SSL yang Anda masukkan pada langkah sebelumnya. Hapus baris tambahan sebelum dan sesudah kunci pribadi. Pilih Berikutnya.
8. Pada halaman Konfigurasi pengaturan lanjutan, di area Jaringan dan keamanan, pilih VPC, subnet, dan satu atau beberapa grup keamanan. AWS OpsWorks dapat menghasilkan grup keamanan, peran layanan, dan profil instance untuk Anda, jika Anda belum memiliki yang ingin Anda gunakan. Server Anda dapat menjadi anggota dari beberapa grup keamanan. Anda tidak dapat mengubah pengaturan jaringan dan keamanan untuk master boneka setelah Anda meninggalkan halaman ini.

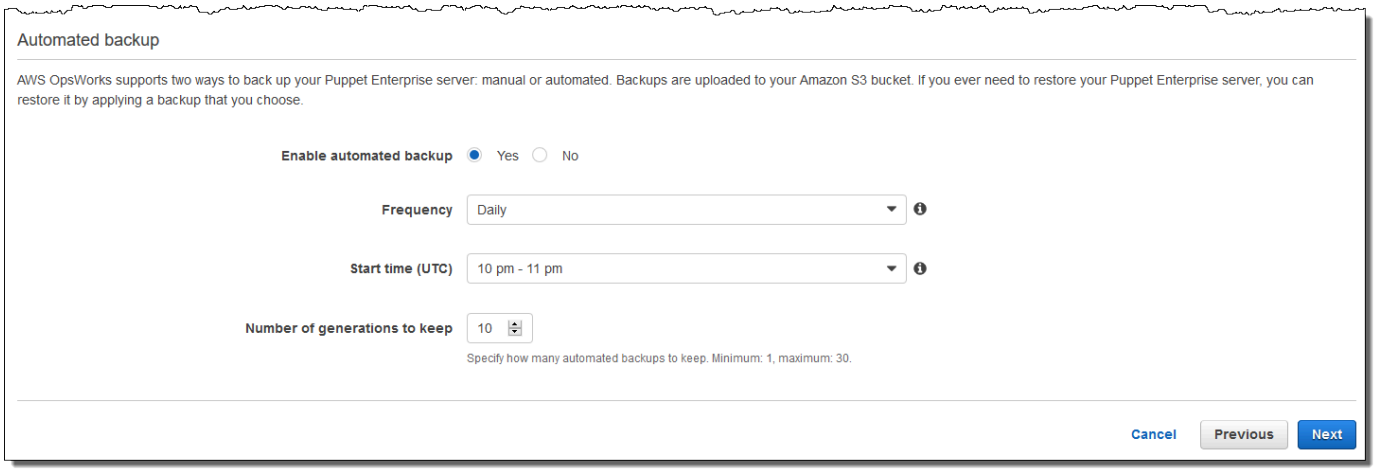


9. Di bagian Pemeliharaan sistem, atur hari dan jam yang Anda inginkan untuk memulai pemeliharaan sistem. Karena Anda harus mengharapkan server offline selama pemeliharaan sistem, pilih waktu permintaan server yang rendah dalam jam kantor biasa.

   Jendela pemeliharaan diperlukan. Anda dapat mengubah hari mulai dan waktu kemudian dengan menggunakan AWS Management Console, AWS CLI, atau APIs.



10. Konfigurasikan cadangan. Secara default, backup otomatis diaktifkan. Tetapkan frekuensi dan jam yang diinginkan agar pencadangan otomatis dimulai, dan atur jumlah generasi cadangan yang akan disimpan di Amazon Simple Storage Service. Maksimal 30 backup dapat disimpan; ketika maksimum tercapai, OpsWorks untuk Puppet Enterprise menghapus backup tertua untuk memberi ruang bagi yang baru.



11. (Opsional) Di Tag, tambahkan tag ke server dan sumber daya terkait, seperti EC2 contoh, alamat IP Elastis, grup keamanan, bucket S3, dan cadangan. Untuk informasi selengkapnya tentang menandai server Puppet Enterprise, lihat. OpsWorks Bekerja dengan Tag pada AWS OpsWorks for Puppet Enterprise Sumber Daya
12. Setelah selesai mengonfigurasi pengaturan lanjutan, pilih Berikutnya.
13. Pada halaman Tinjau, tinjau pilihan Anda. Ketika Anda siap untuk membuat server, pilih Launch.

    Sementara Anda menunggu AWS OpsWorks untuk membuat master Boneka Anda, lanjutkan ke Konfigurasikan Puppet Master Menggunakan Starter Kit dan unduh Starter Kit dan kredenal

konsol Puppet Enterprise. Jangan menunggu sampai server Anda online untuk mengunduh item ini.

Ketika pembuatan server selesai, master Puppet Anda tersedia di halaman beranda OpsWorks for Puppet Enterprise, dengan status online. Setelah server online, konsol Puppet Enterprise tersedia di domain server, di URL dalam format berikut:`https://your_server_name-randomID.region.opsworks-cm.io`.

Buat Master Perusahaan Boneka dengan menggunakan AWS CLI

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Membuat server master OpsWorks for Puppet Enterprise dengan menjalankan AWS CLI perintah berbeda dari membuat server di konsol. Di konsol, AWS OpsWorks buat peran layanan dan grup keamanan untuk Anda, jika Anda tidak menentukan yang sudah ada yang ingin Anda gunakan. Dalam AWS CLI, AWS OpsWorks dapat membuat grup keamanan untuk Anda jika Anda tidak menentukan satu, tetapi tidak secara otomatis membuat peran layanan; Anda harus memberikan peran layanan ARN sebagai bagian dari perintah Anda`create-server`. Di konsol, saat AWS OpsWorks membuat master Boneka Anda, Anda mengunduh starter kit dan kredenal masuk untuk konsol Puppet Enterprise. Karena Anda tidak dapat melakukan ini ketika Anda membuat master OpsWorks for Puppet Enterprise dengan menggunakan AWS CLI, Anda menggunakan utilitas pemrosesan JSON untuk mendapatkan kredensi masuk dan starter kit dari hasil `create-server` perintah setelah master Puppet Enterprise baru Anda OpsWorks online.

Jika komputer lokal Anda belum menjalankan AWS CLI, unduh dan instal AWS CLI dengan mengikuti petunjuk penginstalan di Panduan Pengguna Antarmuka Baris Perintah AWS. Bagian ini tidak menjelaskan semua parameter yang dapat Anda gunakan dengan `create-server` perintah. Untuk informasi selengkapnya tentang `create-server` parameter, lihat `create-server`di AWS CLI Referensi.

1. Pastikan untuk menyelesaikanPrasyarat. Untuk membuat master Puppet Anda, Anda memerlukan ID subnet, jadi Anda harus memiliki VPC.
2. Buat peran layanan dan profil instance.
3. Temukan dan salin ARNs peran layanan di akun Anda.

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

Dalam hasil `list-roles` perintah, cari entri ARN peran layanan yang menyerupai berikut ini. Catat peran layanan ARNs. Anda membutuhkan nilai-nilai ini untuk membuat master Perusahaan Boneka Anda.

```
{
    "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. Temukan dan salin profil instans di akun Anda. ARNs

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

Dalam hasil `list-instance-profiles` perintah, cari contoh entri ARN profil yang menyerupai berikut ini. Buat catatan profil instance ARNs. Anda membutuhkan nilai-nilai ini untuk membuat master Perusahaan Boneka Anda.

```
{
    "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. Buat master OpsWorks for Puppet Enterprise dengan menjalankan `create-server` perintah.

   - `--engine`Nilainya `--engine-model` adalah `PuppetMonolithic`, adalah, dan `--engine-version` bisa `2019` atau`2017`.
   - Nama server harus unik di dalam AWS akun Anda, di setiap wilayah. Nama server harus dimulai dengan huruf; kemudian huruf, angka, atau tanda hubung (-) diperbolehkan, hingga maksimum 40 karakter.
   - Gunakan ARN profil instance dan ARN peran layanan yang Anda salin di Langkah 3 dan 4.
   - Jenis contoh yang valid adalah`m5.xlarge`,`c5.2xlarge`, atau`c5.4xlarge`. Untuk informasi selengkapnya tentang spesifikasi jenis instans ini, lihat Jenis Instance di Panduan EC2 Pengguna Amazon.
   - `--engine-attributes`Parameternya opsional; jika Anda tidak menentukan kata sandi administrator Puppet, proses pembuatan server menghasilkan satu untuk Anda. Jika Anda menambahkan`--engine-attributes`, tentukan`PUPPET_ADMIN_PASSWORD`, kata sandi administrator untuk masuk ke halaman web konsol Puppet Enterprise. Kata sandi harus menggunakan antara 8 hingga 32 karakter ASCII.
   - Sebuah key pair SSH adalah opsional, tetapi dapat membantu Anda terhubung ke master Puppet Anda jika Anda perlu mengatur ulang kata sandi administrator konsol. Untuk informasi selengkapnya tentang membuat key pair SSH, lihat Amazon EC2 Key Pairs di Panduan EC2 Pengguna Amazon.
   - Untuk menggunakan domain khusus, tambahkan parameter berikut ke perintah Anda. Jika tidak, proses pembuatan master Puppet secara otomatis menghasilkan titik akhir untuk Anda. Ketiga parameter diperlukan untuk mengonfigurasi domain khusus. Untuk informasi tentang persyaratan tambahan untuk menggunakan parameter ini, lihat CreateServerdi Referensi AWS OpsWorks CM API.
     - `--custom-domain`- Sebuah endpoint publik opsional dari server, seperti`https://aws.my-company.com`.
     - `--custom-certificate`- Sertifikat HTTPS berformat PEM. Nilai dapat menjadi sertifikat tunggal yang ditandatangani sendiri, atau rantai sertifikat.

  - `--custom-private-key`- Kunci pribadi dalam format PEM untuk menghubungkan ke server dengan menggunakan HTTPS. Kunci privat tidak boleh dienkripsi; tidak dapat dilindungi oleh kata sandi atau frasa sandi.
- Pemeliharaan sistem mingguan diperlukan. Nilai yang valid harus ditentukan dalam format berikut:`DDD:HH:MM`. Waktu yang ditentukan adalah dalam Waktu Universal Terkoordinasi (UTC). Jika Anda tidak menentukan nilai untuk`--preferred-maintenance-window`, nilai default adalah acak, periode satu jam pada hari Selasa, Rabu, atau Jumat.
- Nilai yang valid untuk `--preferred-backup-window` harus ditentukan dalam salah satu format berikut: `HH:MM` untuk backup harian, atau `DDD:HH:MM` untuk backup mingguan. Waktu yang ditentukan adalah dalam UTC. Nilai default adalah waktu mulai acak harian. Untuk memilih keluar dari backup otomatis, tambahkan parameter `--disable-automated-backup` sebagai gantinya.
- Untuk`--security-group-ids`, masukkan satu atau lebih grup keamanan IDs, dipisahkan oleh spasi.
- Untuk`--subnet-ids`, masukkan subnet ID.

```
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
```

Berikut adalah contohnya.

```
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
```

Contoh berikut membuat master Puppet yang menggunakan domain kustom.

```
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
```

Contoh berikut menciptakan master Puppet yang menambahkan dua tag: `Stage: Production` dan`Department: Marketing`. Untuk informasi selengkapnya tentang menambahkan dan mengelola tag OpsWorks untuk server Puppet Enterprise, lihat Bekerja dengan Tag pada AWS OpsWorks for Puppet Enterprise Sumber Daya di panduan ini.

```
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 untuk Puppet Enterprise membutuhkan waktu sekitar 15 menit untuk membuat server baru. Jangan mengabaikan output dari `create-server` perintah atau menutup sesi shell Anda, karena output dapat berisi informasi penting yang tidak ditampilkan lagi. Untuk mendapatkan kata sandi dan starter kit dari `create-server` hasil, lanjutkan ke langkah berikutnya.

   Jika Anda menggunakan domain khusus dengan server, dalam output `create-server` perintah, salin nilai `Endpoint` atribut. Berikut adalah contohnya.

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

7. Jika Anda memilih untuk membuat kata sandi OpsWorks untuk Puppet Enterprise untuk Anda, Anda dapat mengekstraknya dalam format yang dapat digunakan dari `create-server` hasil dengan menggunakan prosesor JSON seperti jq. Setelah Anda menginstal jq, Anda dapat menjalankan perintah berikut untuk mengekstrak kata sandi administrator Puppet dan starter kit. Jika Anda tidak memberikan kata sandi Anda sendiri pada Langkah 3, pastikan untuk menyimpan kata sandi administrator yang diekstraksi di lokasi yang nyaman namun aman.

   ```
   #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**
   >
   > Anda tidak dapat membuat ulang starter kit master Puppet baru di. AWS Management Console Saat Anda membuat master Puppet dengan menggunakan AWS CLI, jalankan `jq` perintah sebelumnya untuk menyimpan starter kit yang dikodekan base64 dalam hasil sebagai file ZIP. `create-server`

8. Jika Anda tidak menggunakan domain khusus, lanjutkan ke langkah berikutnya. Jika Anda menggunakan domain kustom dengan server, buat entri CNAME di alat manajemen DNS

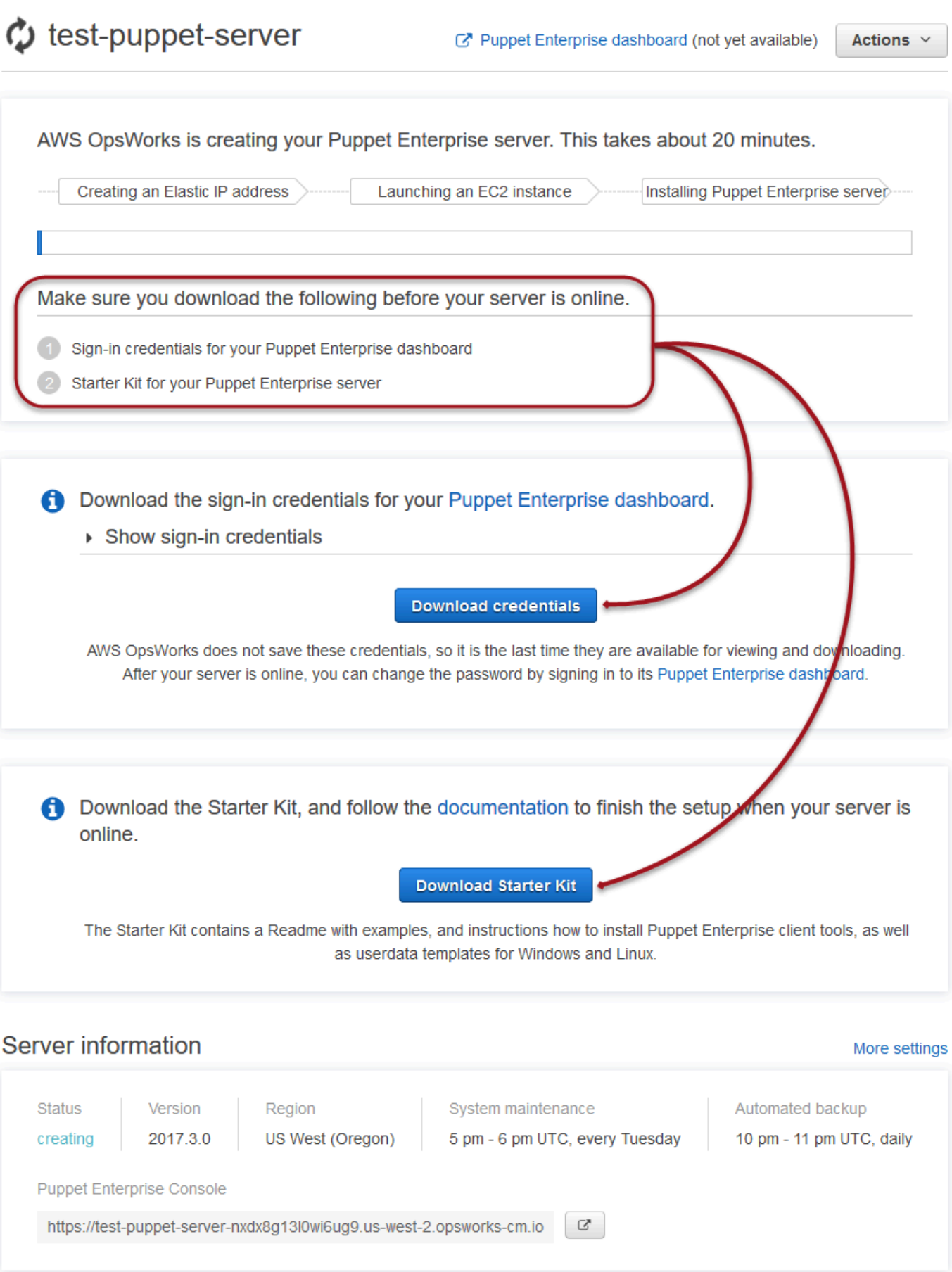
perusahaan Anda untuk mengarahkan domain kustom Anda ke titik akhir OpsWorks untuk Puppet Enterprise yang Anda salin pada langkah 6. Anda tidak dapat menjangkau atau masuk ke server dengan domain khusus sampai Anda menyelesaikan langkah ini.

9. Lanjutkan ke bagian selanjutnya,the section called “Selesai Konfigurasi”.

Konfigurasikan Puppet Master Menggunakan Starter Kit

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sementara pembuatan master boneka masih berlangsung, halaman Properties server terbuka di konsol OpsWorks for Puppet Enterprise. Pertama kali Anda bekerja dengan master Puppet baru, halaman Properties meminta Anda untuk mengunduh dua item yang diperlukan. Unduh item ini sebelum server Puppet Anda online; tombol unduh tidak tersedia setelah server baru online.


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

- Kredensi masuk untuk master Boneka. Anda akan menggunakan kredensil ini untuk masuk ke konsol Puppet Enterprise, tempat Anda melakukan sebagian besar manajemen node. AWS OpsWorks tidak menyimpan kredensil ini; ini adalah terakhir kalinya mereka tersedia untuk dilihat dan diunduh. Jika perlu, Anda dapat mengubah kata sandi yang diberikan dengan kredensil ini setelah Anda masuk.
- Kit Pemula. Starter Kit berisi file README dengan informasi dan contoh yang menjelaskan cara menyelesaikan penyiapan, dan kredensi administrator untuk konsol Puppet Enterprise. Kredensi baru dihasilkan—dan kredensi lama tidak validasi—setiap kali Anda mengunduh Starter Kit.

Prasyarat

1. Sementara pembuatan server masih berlangsung, unduh kredensi masuk untuk master Boneka, dan simpan di lokasi yang aman namun nyaman.
2. Unduh Starter Kit, dan unzip file Starter Kit .zip ke direktori ruang kerja Anda. Jangan bagikan kredensi masuk Anda. Jika pengguna lain akan mengelola master boneka, tambahkan mereka sebagai administrator di konsol Puppet Enterprise nanti. Untuk informasi selengkapnya tentang cara menambahkan pengguna ke master wayang, lihat Membuat dan mengelola pengguna dan peran pengguna dalam dokumentasi Perusahaan Boneka.

Pasang Sertifikat Master Boneka

Untuk bekerja dengan master Puppet Anda dan menambahkan node untuk dikelola, Anda harus menginstal sertifikatnya. Instal dengan menjalankan AWS CLI perintah berikut. Anda tidak dapat melakukan tugas ini di 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
```

Menghasilkan Token Jangka Pendek

Untuk menggunakan API Boneka, Anda harus membuat token jangka pendek untuk diri sendiri. Langkah ini tidak diperlukan untuk menggunakan konsol Puppet Enterprise. Hasilkan token dengan menjalankan perintah berikut.

Masa pakai token default adalah lima menit, tetapi Anda dapat mengubah default ini.

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

> **Note**
>
> Karena masa pakai token default adalah lima menit, perintah contoh sebelumnya menambahkan `--lifetime` parameter untuk memperpanjang masa pakai token untuk periode yang lebih lama. Anda dapat mengatur masa pakai token untuk jangka waktu hingga 10 tahun (`10y`). Untuk informasi selengkapnya tentang cara mengubah masa pakai token default, lihat Mengubah masa pakai default token dalam dokumentasi Puppet Enterprise.

Mengatur Contoh Starter Kit Apache

Setelah mengunduh dan membuka zip Starter Kit, Anda dapat menggunakan cabang contoh di `control-repo-example` folder sampel yang disertakan untuk mengonfigurasi server web Apache di node terkelola Anda.

Starter Kit mencakup dua `control-repo` folder:`control-repo`, dan`control-repo-example`. `control-repo`Folder menyertakan `production` cabang yang tidak berubah dari apa yang akan Anda lihat di repositori Puppet GitHub . `control-repo-example`Folder ini juga memiliki `production` cabang yang menyertakan kode contoh untuk menyiapkan server Apache dengan situs web pengujian.

1. Dorong `control-repo-example` `production` cabang ke remote Git Anda (`r10k_remote`URL master Puppet Anda). Di direktori root Starter Kit Anda, jalankan yang berikut ini, ganti *`r10kRemoteUrl`* dengan `r10k_remote` URL Anda.

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

   Code Manager Puppet menggunakan cabang Git sebagai lingkungan. Secara default, semua node berada di lingkungan produksi.

> **Important**
>
> Jangan mendorong ke `master` cabang. `master`Cabang disediakan untuk master boneka.

2. Menyebarkan kode di `control-repo-example` cabang ke master Puppet Anda. Ini memungkinkan Puppet Master mengunduh kode Puppet Anda dari repositori Git Anda (). `r10k_remote` Di direktori root Starter Kit Anda, jalankan yang berikut ini.

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

Untuk informasi selengkapnya tentang cara menerapkan contoh konfigurasi Apache ke node terkelola yang Anda buat di Amazon EC2, lihat Langkah 2: Buat Instans dengan Menggunakan Skrip Asosiasi Tanpa Pengawasan di panduan ini.

Tambahkan Node untuk Dikelola oleh Puppet Master

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik

- Jalankan panggilan associateNode() API
- Pertimbangan untuk Menambahkan Node Lokal
- Informasi Selengkapnya

Cara yang disarankan untuk menambahkan node adalah dengan menggunakan AWS OpsWorks `associateNode()` API. Server master Puppet Enterprise menghosting repositori yang Anda gunakan untuk menginstal perangkat lunak agen Puppet pada node yang ingin Anda kelola, apakah node adalah komputer fisik lokal atau mesin virtual. Perangkat lunak agen boneka untuk

beberapa sistem operasi diinstal pada server OpsWorks for Puppet Enterprise sebagai bagian dari proses peluncuran. Tabel berikut menunjukkan agen sistem operasi yang tersedia di server Puppet Enterprise Anda OpsWorks saat diluncurkan.

Agen sistem operasi yang sudah diinstal sebelumnya

Sistem operasi yang didukung	Versi
Ubuntu	16.04, 18.04, 20.04
Red Hat Enterprise Linux (RHEL)	6, 7, 8
Windows	Edisi 64-bit dari semua rilis Windows yang didukung Puppet

Anda dapat menambahkan `puppet-agent` ke server Anda untuk sistem operasi lain. Ketahuilah bahwa pemeliharaan sistem akan menghapus agen yang telah Anda tambahkan ke server Anda setelah peluncuran. Meskipun sebagian besar node terlampir yang sudah menjalankan agen yang dihapus terus melakukan check in, node yang menjalankan sistem operasi Debian dapat menghentikan pelaporan. Kami menyarankan Anda menginstal secara manual `puppet-agent` pada node yang menjalankan sistem operasi yang perangkat lunak agen tidak diinstal sebelumnya di server OpsWorks for Puppet Enterprise Anda. Untuk informasi rinci tentang cara membuat `puppet-agent` tersedia di server Anda untuk node dengan sistem operasi lain, lihat Menginstal agen dalam dokumentasi Puppet Enterprise.

Untuk informasi tentang cara mengaitkan node dengan master Puppet Anda secara otomatis dengan mengisi data pengguna EC2 instance, lihat. Menambahkan Node Secara Otomatis OpsWorks untuk Perusahaan Boneka

Jalankan panggilan **`associateNode()`** API

Setelah Anda menambahkan node dengan menginstal`puppet-agent`, node mengirim permintaan penandatanganan sertifikat (CSRs) ke server OpsWorks for Puppet Enterprise. Anda dapat melihat CSRs di konsol Puppet; untuk informasi selengkapnya tentang node CSRs, lihat Mengelola permintaan penandatanganan sertifikat dalam dokumentasi Puppet Enterprise. Menjalankan panggilan OpsWorks for Puppet Enterprise `associateNode()` API memproses node CSRs, dan mengaitkan node dengan server Anda. Berikut ini adalah contoh bagaimana menggunakan panggilan API ini di AWS CLI untuk mengaitkan satu node. Anda akan memerlukan CSR berformat PEM yang dikirim node; Anda bisa mendapatkan ini dari konsol 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'
```

Untuk informasi selengkapnya tentang cara menambahkan node secara otomatis menggunakan`associateNode()`, lihatMenambahkan Node Secara Otomatis OpsWorks untuk Perusahaan Boneka.

Pertimbangan untuk Menambahkan Node Lokal

Setelah Anda menginstal `puppet-agent` di komputer lokal atau mesin virtual, Anda dapat menggunakan salah satu dari dua cara untuk mengaitkan node lokal dengan master OpsWorks for Puppet Enterprise Anda.

- Jika node mendukung penginstalan AWS SDK AWS CLI, atau Alat AWS untuk PowerShell, Anda dapat menggunakan metode yang disarankan untuk mengaitkan node, yaitu menjalankan panggilan `associateNode()` API. Kit starter yang Anda unduh saat pertama kali membuat master OpsWorks for Puppet Enterprise menunjukkan cara menetapkan peran ke node dengan menggunakan tag. Anda dapat menerapkan tag pada saat yang sama ketika Anda mengaitkan node dengan master Puppet dengan menentukan fakta tepercaya di CSR. Misalnya, repositori kontrol demo yang disertakan dengan starter kit dikonfigurasi untuk menggunakan tag untuk menetapkan `pp_role` peran ke instance Amazon. EC2 Untuk informasi selengkapnya tentang cara menambahkan tag ke CSR sebagai fakta tepercaya, lihat Permintaan ekstensi (data sertifikat permanen) di dokumentasi platform Wayang.
- Jika node tidak dapat menjalankan alat AWS manajemen atau pengembangan, Anda masih dapat mendaftarkannya ke master Puppet Enterprise Anda OpsWorks dengan cara yang sama seperti Anda mendaftarkannya ke master Perusahaan Boneka yang tidak dikelola. Seperti disebutkan dalam topik ini, menginstal `puppet-agent` mengirimkan CSR ke master OpsWorks for Puppet Enterprise. Pengguna Puppet resmi dapat menandatangani CSR secara manual, atau mengonfigurasi penandatanganan otomatis CSRs dengan mengedit `autosign.conf` file yang disimpan di master Puppet. Untuk informasi selengkapnya tentang mengonfigurasi penandatanganan otomatis dan pengeditan`autosign.conf`, lihat konfigurasi SSL: permintaan sertifikat penandatanganan otomatis dalam dokumentasi platform Boneka.

Untuk mengaitkan node lokal dengan master Puppet dan mengizinkan master Puppet menerima semua CSRs, lakukan hal berikut di konsol Puppet Enterprise. Parameter yang mengontrol perilaku adalah`puppet_enterprise::profile::master::allow_unauthenticated_ca`.

> **Important**
>
> Mengaktifkan Puppet master untuk menerima tanda tangan sendiri CSRs atau semua tidak CSRs disarankan untuk alasan keamanan. Secara default, mengizinkan yang tidak diautentikasi CSRs membuat master boneka dapat diakses oleh dunia. Mengatur upload permintaan sertifikat untuk diaktifkan secara default dapat membuat master Puppet Anda rentan terhadap serangan denial of service (DoS).

1. Masuk ke konsol Puppet Enterprise.
2. Pilih Konfigurasi, pilih Klasifikasi, pilih Master PE, lalu pilih tab Konfigurasi.
3. Pada tab Klasifikasi, cari kelas puppet_enterprise: :profile: :master.
4. Tetapkan nilai parameter allow_unauthenticated_ca ke true.
5. Simpan perubahan Anda. Perubahan Anda diterapkan selama Puppet run berikutnya. Anda dapat mengizinkan 30 menit agar perubahan diterapkan (dan node lokal ditambahkan), atau Anda dapat memulai Puppet yang dijalankan secara manual di bagian Jalankan konsol PE.

Informasi Selengkapnya

Kunjungi situs tutorial Belajar Boneka untuk mempelajari lebih lanjut tentang penggunaan OpsWorks untuk server Puppet Enterprise dan fitur konsol Puppet Enterprise.

Masuk ke Konsol Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda mengunduh kredensi masuk dari halaman Properti master boneka, dan server sedang online, masuk ke konsol Puppet Enterprise. Dalam panduan ini, kami menginstruksikan Anda untuk menentukan repositori kontrol Anda yang berisi modul Anda, dan menambahkan setidaknya satu

node untuk dikelola. Ini memungkinkan Anda untuk melihat informasi tentang agen dan node di konsol.

Ketika Anda mencoba untuk terhubung ke halaman web konsol Puppet Enterprise, peringatan sertifikat muncul di browser Anda sampai Anda menginstal sertifikat SSL AWS OpsWorks khusus yang ditandatangani CA pada komputer klien yang Anda gunakan untuk mengelola server Puppet Anda. Jika Anda memilih untuk tidak melihat peringatan sebelum melanjutkan ke halaman web dasbor, instal sertifikat SSL sebelum Anda masuk.

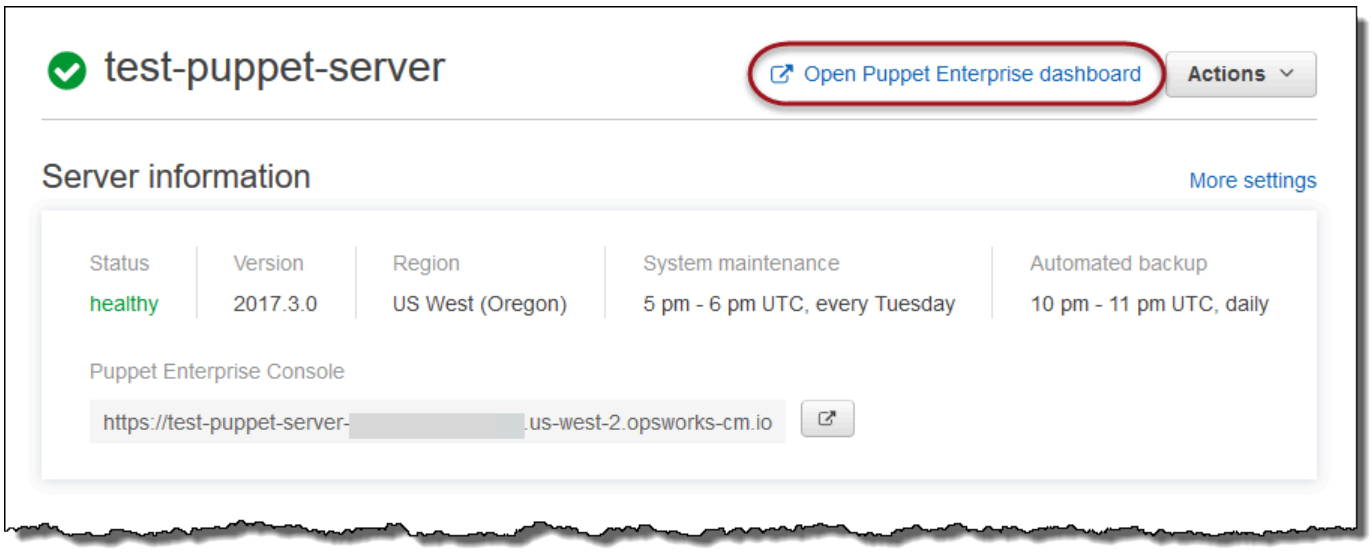
Untuk menginstal sertifikat AWS OpsWorks SSL

- Pilih sertifikat yang sesuai dengan sistem Anda.

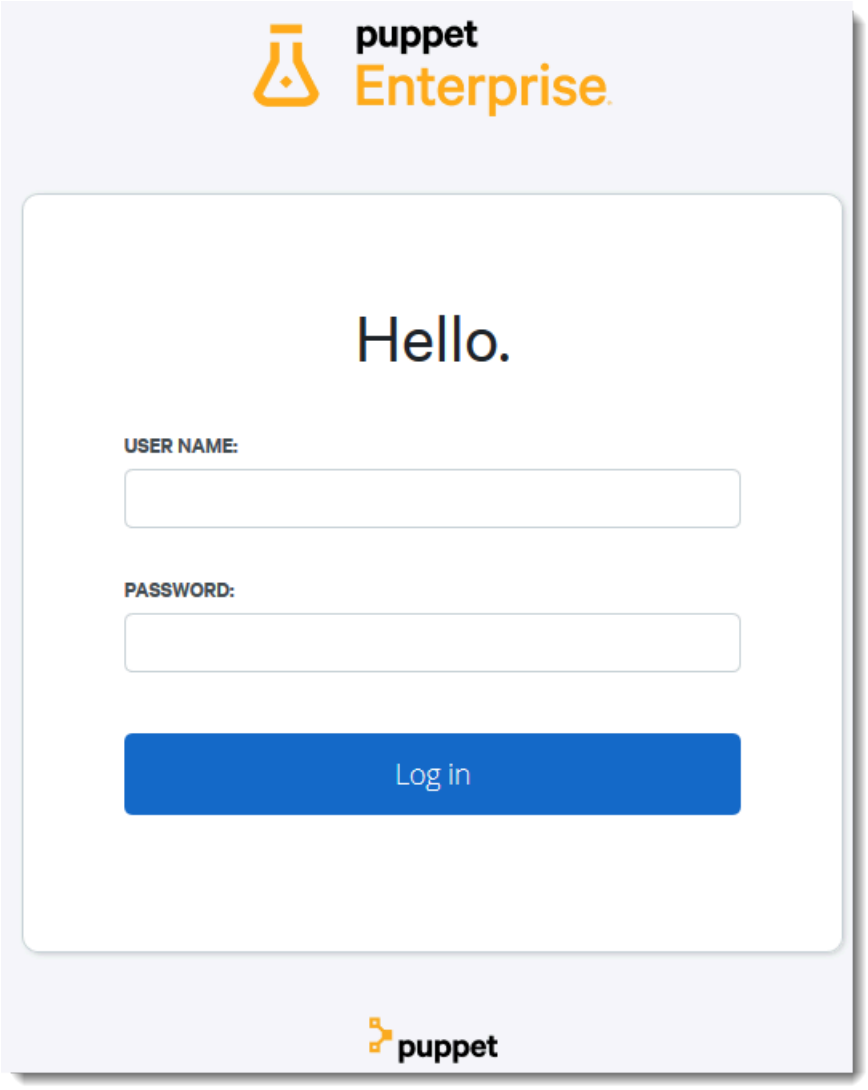
Setelah Anda menginstal sertifikat SSL sisi klien, Anda dapat masuk ke konsol Puppet Enterprise tanpa melihat pesan peringatan.

Untuk masuk ke konsol Puppet Enterprise

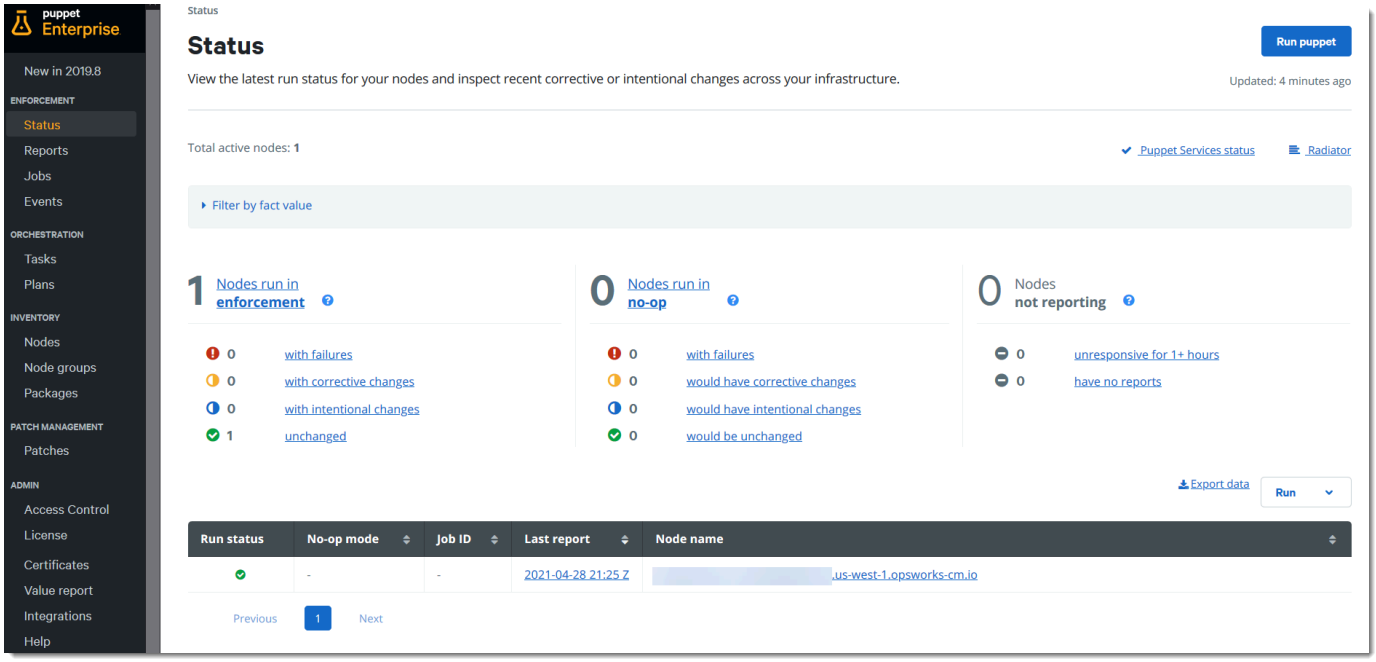
1. Buka zip dan buka kredenal Puppet Enterprise yang Anda unduh. Prasyarat Anda akan memerlukan kredensi ini untuk masuk.
2. Di AWS Management Console, buka halaman Properties untuk server Puppet Anda.
3. Di kanan atas halaman Properties, pilih Open Puppet Enterprise console.



4. Masuk menggunakan kredensil dari Langkah 1.



5. Di konsol Puppet Enterprise, Anda dapat melihat informasi terperinci tentang node yang Anda kelola, progres dan peristiwa yang dijalankan modul, tingkat kepatuhan node, dan banyak lagi. Untuk informasi selengkapnya tentang fitur konsol Puppet Enterprise dan cara menggunakannya, lihat Mengelola node dalam dokumentasi Puppet Enterprise.



Kelompokkan dan Klasifikasi Node

Sebelum Anda menentukan konfigurasi node yang diinginkan dengan menerapkan kelas ke node tersebut, kelompokkan node sesuai dengan perannya di perusahaan Anda atau karakteristik umum mereka. Pengelompokan dan klasifikasi node melibatkan tugas-tugas tingkat tinggi berikut. Anda dapat menyelesaikan tugas-tugas ini dengan menggunakan konsol PE. Untuk informasi rinci tentang cara mengelompokkan dan mengklasifikasikan node Anda, lihat Mengelompokkan dan mengklasifikasikan node dalam dokumentasi Puppet Enterprise.

1. Buat grup simpul.
2. Tambahkan node ke grup secara manual atau otomatis dengan menerapkan aturan yang Anda buat.
3. Tetapkan kelas ke grup simpul.

Setel Ulang Administrator dan Kata Sandi Pengguna

Untuk informasi tentang cara mengubah kata sandi yang Anda gunakan untuk masuk ke konsol Puppet Enterprise, lihat Mengatur ulang kata sandi administrator konsol dalam dokumentasi Perusahaan Boneka.

Secara default, setelah sepuluh upaya masuk, pengguna dikunci dari konsol Puppet. Untuk informasi selengkapnya tentang cara mengatur ulang kata sandi pengguna jika terjadi penguncian, lihat Titik akhir kata sandi dalam dokumentasi Perusahaan Boneka.

Opsional: Gunakan AWS CodeCommit sebagai Repositori Remote Control Boneka r10k

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

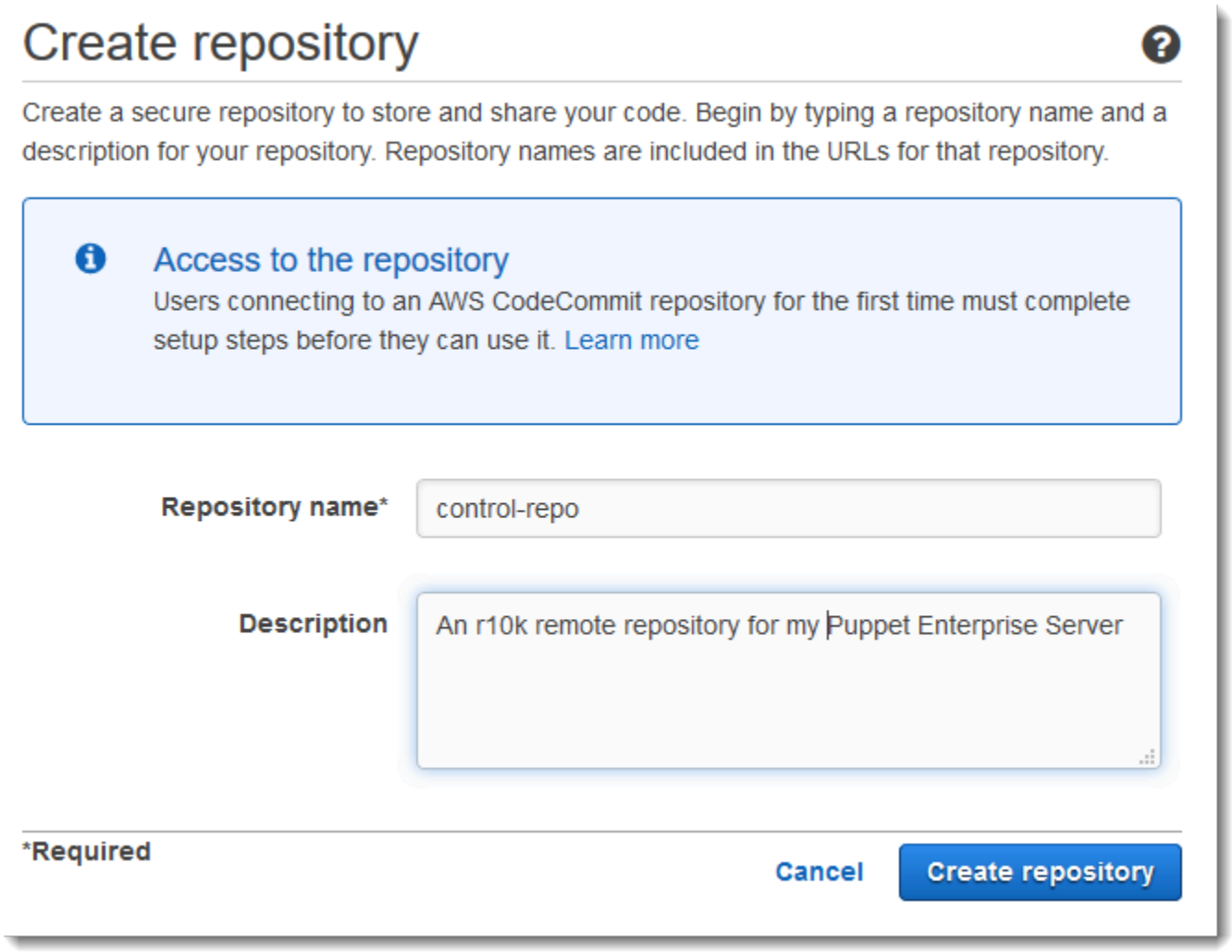
Anda dapat membuat repositori baru dengan menggunakan AWS CodeCommit, dan menggunakannya sebagai repositori remote control r10k Anda. Untuk menyelesaikan langkah-langkah di bagian ini, dan bekerja dengan CodeCommit repositori, Anda memerlukan pengguna yang memiliki izin yang disediakan oleh kebijakan AWSCodeCommitReadOnlyterkelola.

Topik

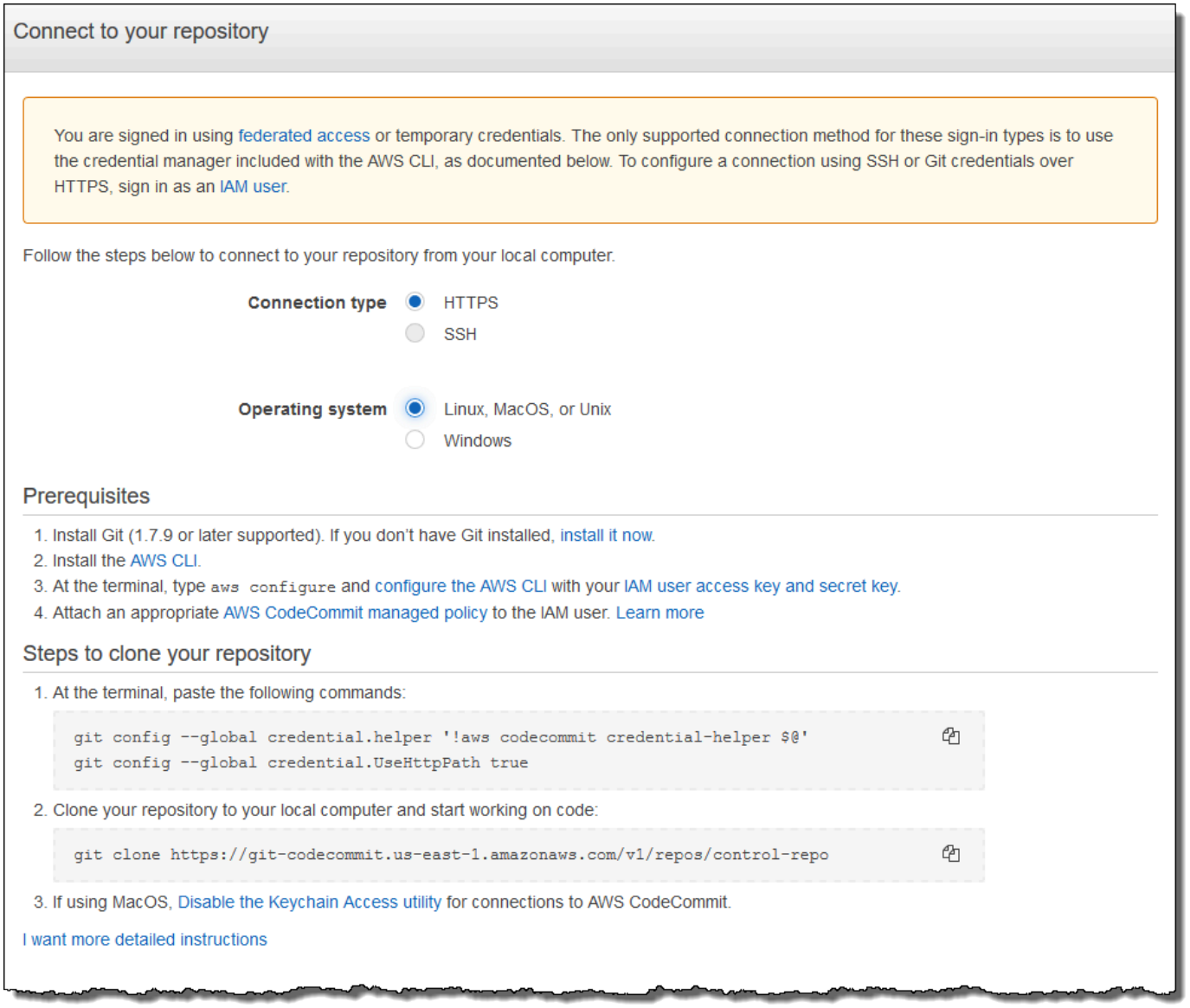
- Langkah 1: Gunakan CodeCommit sebagai Repositori dengan tipe koneksi HTTPS
- Langkah 2: (Opsional) Gunakan CodeCommit sebagai Repositori dengan tipe koneksi SSH

Langkah 1: Gunakan CodeCommit sebagai Repositori dengan tipe koneksi HTTPS

1. Di CodeCommit konsol, buat repositori baru.



2. Pilih Lewati untuk melewati pengaturan topik Amazon SNS.
3. Pada halaman Kode, pilih Connect to your repository.
4. Pada halaman Connect to your repository, pilih HTTPS sebagai tipe Connection, dan pilih sistem operasi Anda.



Di Langkah-langkah untuk mengkloning area repositori Anda, `git clone` URL Anda harus menyerupai yang berikut:. `https://git-codecommit.`*`region`*`.amazonaws.com/v1/repos/control-repo` Salin URL ini ke tempat yang nyaman untuk digunakan dalam pengaturan server Puppet.

5. Tutup halaman Connect to repository Anda, dan kembali ke setup server OpsWorks for Puppet Enterprise.
6. Tempel URL yang Anda salin di Langkah 4 di kotak string jarak jauh r10k di halaman Konfigurasi kredensial dari wizard pengaturan master boneka. Biarkan kotak kunci pribadi r10k kosong. Selesai membuat dan meluncurkan master boneka Anda.

7. Di konsol IAM, lampirkan AWSCodeCommitReadOnlykebijakan ke peran profil instance master Puppet Anda. Untuk informasi selengkapnya tentang cara menambahkan kebijakan izin ke peran IAM, lihat Menambahkan izin identitas (konsol) IAM di Panduan Pengguna IAM.
8. Ikuti langkah-langkah dalam Pengaturan untuk pengguna HTTPS yang menggunakan kredensi Git di Panduan AWS CodeCommit Pengguna untuk mendorong `control-repo` konten Anda yang sudah ada ke repositori baru CodeCommit .
9. Sekarang, Anda dapat melanjutkan dengan mengikuti instruksi dithe section called “Selesai Konfigurasi”, dan menggunakan Starter Kit untuk menyebarkan kode ke master Puppet Anda. Berikut adalah contoh perintah tersebut.

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

Langkah 2: (Opsional) Gunakan CodeCommit sebagai Repositori dengan tipe koneksi SSH

Anda dapat mengkonfigurasi repositori remote control AWS CodeCommit r10k untuk menggunakan otentikasi key pair SSH. Prasyarat berikut harus diselesaikan sebelum Anda memulai prosedur ini.

- Anda harus telah meluncurkan server OpsWorks for Puppet Enterprise Anda dengan repositori kontrol HTTPS seperti yang dijelaskan di bagian sebelumnya,. the section called “Langkah 1: Gunakan CodeCommit sebagai Repositori dengan tipe koneksi HTTPS” Ini harus diselesaikan terlebih dahulu sehingga Anda dapat mengunggah konfigurasi yang diperlukan ke master Wayang.
- Pastikan Anda memiliki pengguna dengan kebijakan AWSCodeCommitReadOnlyterkelola terlampir. Untuk informasi selengkapnya tentang cara membuat pengguna, lihat Membuat pengguna IAM di AWS akun Anda di Panduan Pengguna IAM.
- Buat dan kaitkan kunci SSH dengan pengguna Anda. Ikuti petunjuk untuk membuat public/private key pair dengan *`ssh-keygen`* Langkah 3: Konfigurasikan kredenal di Linux, macOS, atau Unix di Panduan Pengguna.AWS CodeCommit

1. Dalam AWS CLI sesi, jalankan perintah berikut untuk mengunggah konten file kunci pribadi ke AWS Systems Manager Parameter Store. Server OpsWorks for Puppet Enterprise Anda menanyakan parameter ini untuk mendapatkan file sertifikat yang diperlukan. Ganti *`private_key_file`* dengan path ke file kunci pribadi SSH Anda.

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

2. Tambahkan izin Systems Manager Parameter Store ke master Puppet Anda.

   a. Buka konsol IAM di https://console.aws.amazon.com/iam/.

   b. Di panel navigasi sebelah kiri, pilih Peran.

   c. Pilih aws-opsworks-cm-ec2-peran.

   d. Di tab Izin, pilih Lampirkan kebijakan.

   e. Di bilah Pencarian, masukkan**AmazonSSMManagedInstanceCore**.

   f. Dalam hasil pencarian, pilih Amazon SSMManaged InstanceCore.

   g. Pilih Lampirkan kebijakan.

3. Buat manifes file konfigurasi. Jika Anda menggunakan `control-repo-example` repositori yang disediakan dalam starter kit, buat file berikut di lokasi yang ditunjukkan dalam contoh repositori. Jika tidak, buat sesuai dengan struktur repositori kontrol Anda sendiri. Ganti *IAM_USER_SSH_KEY* nilai dengan ID kunci SSH yang Anda buat dalam prasyarat untuk prosedur ini.

```
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. Dorong repositori kontrol Anda ke. CodeCommit Jalankan perintah berikut untuk mendorong file manifes baru ke repositori Anda.

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

5. Menyebarkan file manifes. Jalankan perintah berikut untuk menyebarkan konfigurasi yang diperbarui ke server OpsWorks for Puppet Enterprise Anda. Ganti *STARTER_KIT_DIRECTORY* dengan path ke file konfigurasi Puppet Anda.

```
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. Perbarui klasifikasi server OpsWorks untuk Puppet Enterprise. Secara default, agen Puppet berjalan pada node (termasuk master) setiap 30 menit. Untuk menghindari menunggu, Anda dapat menjalankan agen secara manual pada master boneka. Menjalankan agen mengambil file manifes baru.

   a. Masuk ke konsol Puppet Enterprise.

   b. Pilih Klasifikasi.

   c. Perluas Infrastruktur PE.

   d. Pilih PE Master.

   e. Pada tab Konfigurasi, **`profile::codecommit`** masukkan Tambah kelas baru.

      Kelas baru,`profile::codecommit`, mungkin tidak muncul segera setelah dijalankan`puppet-code deploy`. Pilih Refresh di halaman ini jika tidak muncul.

   f. Pilih Tambah kelas, lalu pilih Commit 1 change.

   g. Jalankan agen Puppet secara manual di server OpsWorks for Puppet Enterprise. Pilih Node, pilih server Anda dalam daftar, pilih Jalankan Boneka, lalu pilih Jalankan.

7. Di konsol Puppet Enterprise, ubah URL repositori untuk menggunakan SSH, bukan HTTPS. Konfigurasi yang Anda lakukan dalam langkah-langkah ini disimpan selama proses pencadangan dan pemulihan OpsWorks for Puppet Enterprise, sehingga Anda tidak perlu mengubah konfigurasi repositori secara manual setelah aktivitas pemeliharaan.

   a. Pilih Klasifikasi.

   b. Perluas Infrastruktur PE.

   c. Pilih PE Master.

d. Pada tab Konfigurasi, temukan `puppet_enterprise::profile::master` kelas.

e. Pilih Edit di sebelah `r10k_remote` parameter.

f. Ganti URL HTTPS dengan URL SSH untuk repositori Anda, lalu pilih Commit 1 change.

g. Jalankan agen Puppet secara manual di server OpsWorks for Puppet Enterprise. Pilih Node, pilih server Anda dalam daftar, pilih Jalankan Boneka, lalu pilih Jalankan.

Buat AWS OpsWorks for Puppet Enterprise Master dengan menggunakan AWS CloudFormation

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks for Puppet Enterprise memungkinkan Anda menjalankan server Puppet Enterprise di AWS. Anda dapat menyediakan server master Puppet Enterprise dalam waktu sekitar 15 menit.

Mulai 3 Mei 2021, OpsWorks untuk Puppet Enterprise menyimpan beberapa atribut server Puppet Enterprise di. AWS Secrets Manager Untuk informasi selengkapnya, lihat Integrasi dengan AWS Secrets Manager.

Panduan berikut membantu Anda membuat master Boneka OpsWorks untuk Perusahaan Boneka dengan membuat tumpukan. AWS CloudFormation

Topik

- Prasyarat
- Buat Master Perusahaan Boneka di AWS CloudFormation

Prasyarat

Sebelum Anda membuat master Wayang baru, buat sumber daya di luar OpsWorks untuk Perusahaan Boneka yang harus Anda akses dan kelola master Wayang Anda. Untuk informasi selengkapnya, lihat Prasyarat di bagian Memulai panduan ini.

Jika Anda membuat server yang menggunakan domain khusus, Anda memerlukan domain kustom, sertifikat, dan kunci pribadi. Anda harus menentukan nilai untuk ketiga parameter ini di AWS CloudFormation template Anda. Untuk informasi selengkapnya tentang persyaratan untuk`CustomDomain`,`CustomCertificate`, dan `CustomPrivateKey` parameter, lihat CreateServerdi Referensi AWS OpsWorks CM API.

Tinjau bagian OpsWorks -CM dari Referensi Template Panduan AWS CloudFormation Pengguna untuk mempelajari nilai yang didukung dan yang diperlukan dalam AWS CloudFormation templat yang Anda gunakan untuk membuat server.

Buat Master Perusahaan Boneka di AWS CloudFormation

Bagian ini menjelaskan cara menggunakan AWS CloudFormation template untuk membangun tumpukan yang membuat server master OpsWorks untuk Puppet Enterprise. Anda dapat melakukan ini dengan menggunakan AWS CloudFormation konsol atau AWS CLI. Contoh AWS CloudFormation template tersedia untuk Anda gunakan untuk membangun tumpukan server OpsWorks untuk Puppet Enterprise. Pastikan untuk memperbarui template contoh dengan nama server Anda sendiri, peran IAM, profil instance, deskripsi server, jumlah retensi cadangan, opsi pemeliharaan, dan tag opsional. Jika server Anda akan menggunakan domain kustom, Anda harus menentukan nilai untuk`CustomDomain`,`CustomCertificate`, dan `CustomPrivateKey` parameter dalam AWS CloudFormation template Anda. Untuk informasi selengkapnya tentang opsi ini, lihat the section called “Buat Master Perusahaan Boneka dengan menggunakan AWS Management Console” di bagian Memulai panduan ini.

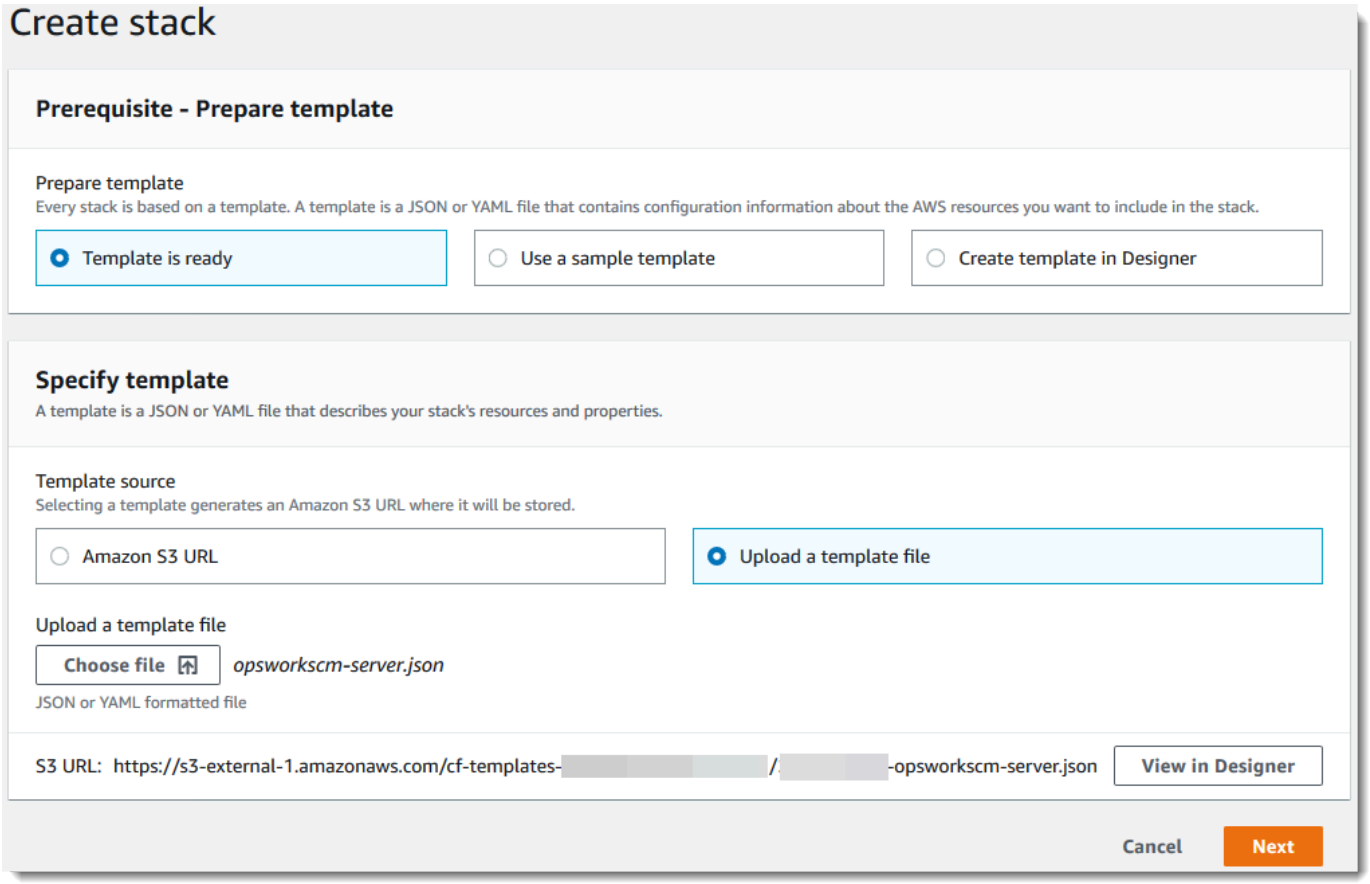
Topik

- Buat Master Perusahaan Boneka dengan menggunakan AWS CloudFormation (Konsol)
- Buat Master Perusahaan Boneka dengan menggunakan AWS CloudFormation (CLI)

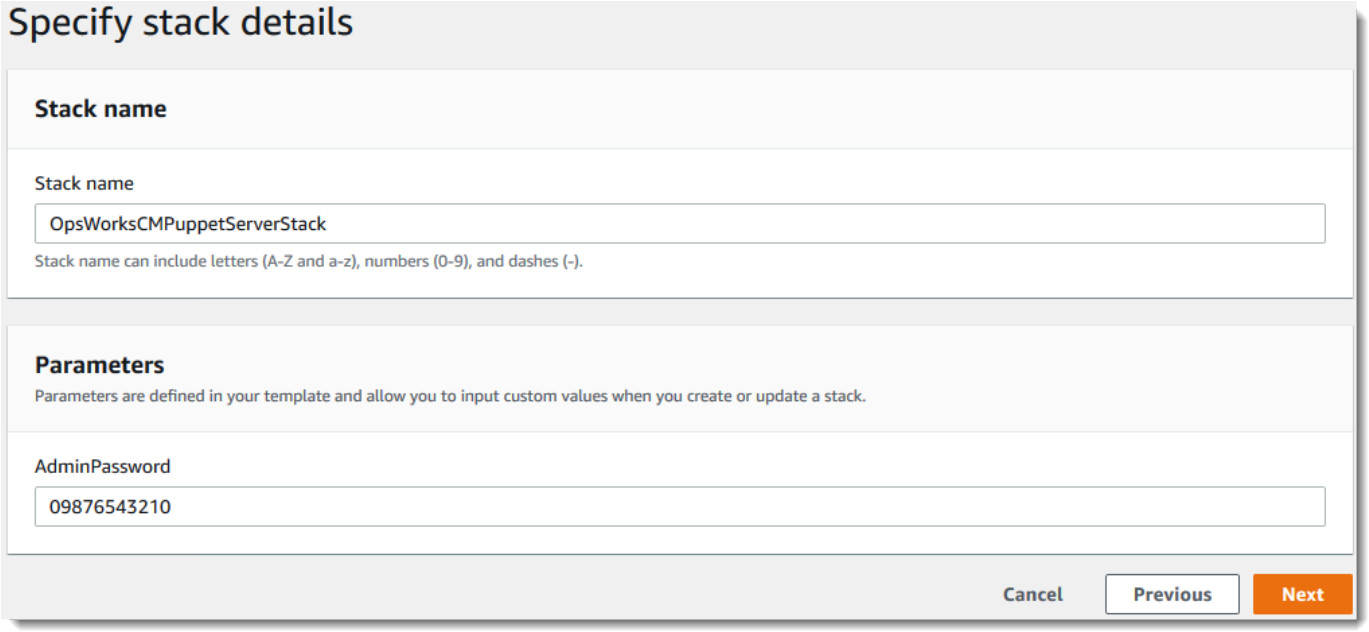
Buat Master Perusahaan Boneka dengan menggunakan AWS CloudFormation (Konsol)

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di AWS CloudFormation halaman beranda, pilih Buat tumpukan.
3. Dalam Prasyarat - Siapkan template, jika Anda menggunakan contoh AWS CloudFormation template, pilih Template siap.
4. Di Tentukan template, pilih sumber template Anda. Untuk panduan ini, pilih Upload file template, dan upload AWS CloudFormation template yang membuat server Puppet Enterprise. Jelajahi file template Anda, lalu pilih Berikutnya.

   AWS CloudFormation Template dapat dalam format YAMAL atau JSON. Contoh AWS CloudFormation template tersedia untuk Anda gunakan; pastikan untuk mengganti nilai contoh dengan milik Anda sendiri. Anda dapat menggunakan desainer AWS CloudFormation template untuk membangun template baru atau memvalidasi yang sudah ada. Untuk informasi selengkapnya tentang cara melakukannya, lihat Ikhtisar Antarmuka AWS CloudFormation Desainer di Panduan AWS CloudFormation Pengguna.



5. Pada halaman Tentukan detail tumpukan, masukkan nama untuk tumpukan Anda. Ini tidak akan sama dengan nama server Anda, itu hanya nama tumpukan. Di area Parameter, masukkan kata sandi administrator untuk masuk ke halaman web konsol Perusahaan Boneka. Kata sandi harus menggunakan antara 8 hingga 32 karakter ASCII. Pilih Berikutnya.



6. Pada halaman Opsi, Anda dapat menambahkan tag ke server yang Anda buat dengan tumpukan, dan memilih peran IAM untuk membuat sumber daya jika Anda belum menentukan peran IAM untuk digunakan dalam template Anda. Setelah selesai menentukan opsi, pilih Berikutnya. Untuk informasi selengkapnya tentang opsi lanjutan seperti pemicu rollback, lihat Mengatur Opsi AWS CloudFormation Tumpukan di Panduan Pengguna.AWS CloudFormation
7. Pada halaman Tinjau, tinjau pilihan Anda. Saat Anda siap membuat tumpukan server, pilih Buat.

   Saat Anda menunggu AWS CloudFormation untuk membuat tumpukan, lihat status pembuatan tumpukan. Jika pembuatan tumpukan gagal, tinjau pesan kesalahan yang ditampilkan di konsol untuk membantu Anda menyelesaikan masalah. Untuk informasi selengkapnya tentang pemecahan masalah kesalahan di AWS CloudFormation tumpukan, lihat Memecahkan Masalah Kesalahan di Panduan Pengguna.AWS CloudFormation

   Ketika pembuatan server selesai, master Puppet Anda tersedia di halaman beranda OpsWorks for Puppet Enterprise, dengan status online. Setelah server online, konsol Puppet Enterprise tersedia di domain server, di URL dalam format berikut:`https://`*`your_server_name`*`-`*`randomID`*`.`*`region`*`.opsworks-cm.io`.

   > **Note**
   >
   > Jika Anda menetapkan domain kustom, sertifikat, dan kunci pribadi untuk server Anda, buat entri CNAME di alat manajemen DNS perusahaan Anda yang memetakan domain kustom Anda ke titik akhir yang dibuat secara otomatis OpsWorks untuk Puppet

Enterprise untuk server. Anda tidak dapat mengelola server atau terhubung ke situs web manajemen Puppet Enterprise untuk server hingga Anda memetakan titik akhir yang dihasilkan ke nilai domain kustom Anda.
Untuk mendapatkan nilai endpoint yang dihasilkan, jalankan AWS CLI perintah berikut setelah server Anda online:

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

Buat Master Perusahaan Boneka dengan menggunakan AWS CloudFormation (CLI)

Jika komputer lokal Anda belum menjalankan AWS CLI, unduh dan instal AWS CLI dengan mengikuti petunjuk penginstalan di Panduan Pengguna Antarmuka Baris Perintah AWS. Bagian ini tidak menjelaskan semua parameter yang dapat Anda gunakan dengan create-stack perintah. Untuk informasi selengkapnya tentang create-stack parameter, lihat create-stackdi AWS CLI Referensi.

1. Pastikan untuk menyelesaikan untuk membuat master Prasyarat OpsWorks for Puppet Enterprise.
2. Buat peran layanan dan profil instance.

   Temukan dan salin ARNs peran layanan di akun Anda.

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

   Dalam hasil `list-roles` perintah, cari peran layanan dan entri profil instance yang menyerupai berikut ini. Catat peran layanan dan profil instance, dan tambahkan ke AWS CloudFormation template yang Anda gunakan untuk membuat tumpukan server master Puppet Anda. ARNs

   ```
   {
       "AssumeRolePolicyDocument": {
           "Version": "2012-10-17",
           "Statement": [
               {
                   "Action": "sts:AssumeRole",
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "ec2.amazonaws.com"
                   }
               }
   ```

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

3. Buat master OpsWorks for Puppet Enterprise dengan menjalankan create-stack perintah lagi.

   - Ganti *`stack_name`* dengan nama tumpukan Anda. Ini adalah nama AWS CloudFormation tumpukan, bukan master boneka Anda. Nama master boneka adalah nilai `ServerName` dalam AWS CloudFormation template Anda.
   - Ganti *`template`* dengan path ke file template Anda, dan ekstensi *`yaml or json`* dengan `.yaml` atau yang `.json` sesuai.
   - Nilai untuk `--parameters` sesuai dengan EngineAttributesdari CreateServerAPI. Untuk Puppet, berikut ini adalah atribut mesin yang disediakan pengguna untuk membuat server. Atribut mesin r10k menghubungkan master Puppet Anda ke repositori kode untuk mengelola konfigurasi lingkungan server. Untuk informasi selengkapnya tentang atribut mesin r10k, lihat Mengelola kode dengan r10k di dokumentasi Puppet Enterprise.

- PUPPET_ADMIN_PASSWORD, kata sandi administrator untuk masuk ke halaman web konsol Puppet Enterprise. Kata sandi harus menggunakan antara 8 dan 32 karakter ASCII, dan memerlukan setidaknya satu huruf besar, satu huruf kecil, satu angka, dan satu karakter khusus.
- PUPPET_R10K_REMOTE, URL repositori kontrol Anda (misalnya, ssh://git@your.git-repo.com:user/control-repo.git). Menentukan remote r10k akan membuka port TCP 8170.
- PUPPET_R10K_PRIVATE_KEY. Jika Anda menggunakan repositori Git pribadi, tambahkan PUPPET_R10K_PRIVATE_KEY untuk menentukan URL SSH dan kunci SSH pribadi yang dikodekan PEM.

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

Berikut adalah contohnya.

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

Contoh berikut menentukan atribut mesin r10k sebagai parameter, ketika mereka tidak disediakan dalam template. AWS CloudFormation Contoh template yang menyertakan atribut mesin r10k`puppet-server-param-attributes.yaml`, disertakan dalam contoh AWS CloudFormation templat.

```
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)"
```

Contoh berikut menentukan atribut mesin r10k dan nilainya dalam AWS CloudFormation template; perintah hanya perlu menunjuk ke file template. Template yang ditentukan sebagai nilai `--template-body`puppet-server-in-file-attributes.yaml,, disertakan dalam contoh AWS CloudFormation template.

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

4. (Opsional) Untuk mendapatkan status pembuatan tumpukan, jalankan perintah berikut.

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

5. Ketika pembuatan tumpukan telah selesai, lanjutkan ke bagian berikutnya,the section called "Selesai Konfigurasi". Jika pembuatan tumpukan gagal, tinjau pesan kesalahan yang ditampilkan di konsol untuk membantu Anda menyelesaikan masalah. Untuk informasi selengkapnya tentang pemecahan masalah kesalahan di AWS CloudFormation tumpukan, lihat Memecahkan Masalah Kesalahan di Panduan Pengguna.AWS CloudFormation

Memperbarui Server Perusahaan Boneka OpsWorks untuk Menggunakan Domain Kustom

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara memperbarui server Puppet Enterprise yang sudah ada OpsWorks untuk menggunakan domain dan sertifikat khusus dengan menggunakan cadangan server untuk membuat server baru. Pada dasarnya, Anda menyalin yang sudah ada OpsWorks untuk server Puppet Enterprise 2.0 dengan membuat server baru dari cadangan, kemudian mengonfigurasi server baru untuk menggunakan domain kustom, sertifikat, dan kunci pribadi.

Topik

- Prasyarat
- Batasan
- Memperbarui Server untuk Menggunakan Domain Kustom

- Lihat Juga

Prasyarat

Berikut ini adalah persyaratan untuk memperbarui server Puppet Enterprise yang ada OpsWorks untuk menggunakan domain dan sertifikat khusus.

- Server yang ingin Anda perbarui (atau salin) harus menjalankan Puppet Enterprise 2019.8.5.
- Tentukan cadangan mana yang ingin Anda gunakan untuk membuat server baru. Anda harus memiliki setidaknya satu cadangan yang tersedia dari server yang ingin Anda perbarui. Untuk informasi lebih lanjut tentang backup OpsWorks untuk Puppet Enterprise, lihat. Cadangkan OpsWorks untuk Server Perusahaan Boneka
- Siapkan peran layanan dan profil instance ARNs yang Anda gunakan untuk membuat server yang ada yang merupakan sumber cadangan Anda.
- Pastikan bahwa Anda menjalankan rilis terbaru dari AWS CLI. Untuk informasi selengkapnya tentang memperbarui AWS CLI alat Anda, lihat Menginstal AWS CLI di Panduan Pengguna Antarmuka Baris Perintah AWS.

Batasan

Ketika Anda memperbarui server yang ada dengan membuat server baru dari cadangan, server baru tidak dapat persis sama dengan yang ada OpsWorks untuk server Puppet Enterprise.

- Anda hanya dapat menyelesaikan prosedur ini dengan menggunakan AWS CLI atau salah satu dari AWS SDKs. Anda tidak dapat membuat server baru dari cadangan dengan menggunakan file AWS Management Console.
- Server baru tidak dapat menggunakan nama yang sama dengan server yang ada di dalam akun, dan dalam Wilayah AWS. Nama harus berbeda dari server yang ada yang Anda gunakan sebagai sumber cadangan.
- Node yang dilampirkan ke server yang ada tidak dikelola oleh server baru. Anda harus melakukan salah satu dari yang berikut ini.
  - Lampirkan node yang berbeda, karena node tidak dapat dikelola oleh lebih dari satu master boneka.

- Migrasikan node dari server yang ada (sumber cadangan) ke server baru dan endpoint domain kustom baru. Untuk informasi selengkapnya tentang cara memigrasi node, lihat dokumentasi Puppet Enterprise.

Memperbarui Server untuk Menggunakan Domain Kustom

Untuk memperbarui master Puppet yang ada, Anda membuat salinannya dengan menjalankan `create-server` perintah, menambahkan parameter untuk menentukan cadangan, domain khusus, sertifikat khusus, dan kunci pribadi khusus.

1. Jika Anda tidak memiliki peran layanan atau profil instans yang ARNs tersedia untuk ditentukan dalam `create-server` perintah Anda, ikuti langkah 1-5 Buat server Chef Automate dengan menggunakan AWS CLI untuk membuat peran layanan dan profil instance yang dapat Anda gunakan.
2. Jika Anda belum melakukannya, temukan cadangan master boneka yang ada di mana Anda ingin mendasarkan server baru dengan domain khusus. Jalankan perintah berikut untuk menampilkan informasi tentang semua OpsWorks cadangan Puppet Enterprise di akun Anda, dan di suatu wilayah. Pastikan untuk mencatat ID cadangan yang ingin Anda gunakan.

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

3. Buat server OpsWorks for Puppet Enterprise dengan menjalankan `create-server` perintah.

   - `--engine`Nilai adalah`Puppet`, `--engine-model` adalah`Monolithic`, dan `--engine-version` adalah `2019` atau`2017`.
   - Nama server harus unik di dalam AWS akun Anda, di setiap wilayah. Nama server harus dimulai dengan huruf; kemudian huruf, angka, atau tanda hubung (-) diperbolehkan, hingga maksimum 40 karakter.
   - Gunakan ARN profil instance dan ARN peran layanan yang Anda salin di Langkah 3 dan 4.
   - Jenis contoh yang valid adalah`c4.large`,`c4.xlarge`, atau`c4.2xlarge`. Untuk informasi selengkapnya tentang spesifikasi jenis instans ini, lihat Jenis Instance di Panduan EC2 Pengguna Amazon.
   - `--engine-attributes`Parameternya opsional; jika Anda tidak menentukan kata sandi administrator Puppet, proses pembuatan server menghasilkan satu untuk Anda. Jika Anda menambahkan`--engine-attributes`, tentukan`PUPPET_ADMIN_PASSWORD`, kata sandi

administrator untuk masuk ke halaman web konsol Puppet Enterprise. Kata sandi harus menggunakan antara 8 hingga 32 karakter ASCII.

- Sebuah key pair SSH adalah opsional, tetapi dapat membantu Anda terhubung ke master Puppet Anda jika Anda perlu mengatur ulang kata sandi administrator konsol. Untuk informasi selengkapnya tentang membuat key pair SSH, lihat Amazon EC2 Key Pairs di Panduan EC2 Pengguna Amazon.
- Untuk menggunakan domain kustom, tambahkan parameter berikut ke perintah Anda. Jika tidak, proses pembuatan master Puppet secara otomatis menghasilkan titik akhir untuk Anda. Ketiga parameter tersebut diperlukan untuk mengonfigurasi domain khusus. Untuk informasi tentang persyaratan tambahan untuk menggunakan parameter ini, lihat CreateServerdi Referensi AWS OpsWorks CM API.
  - `--custom-domain`- Sebuah endpoint publik opsional dari server, sepertihttps://aws.my-company.com.
  - `--custom-certificate`- Sertifikat HTTPS berformat PEM. Nilai dapat menjadi sertifikat tunggal yang ditandatangani sendiri, atau rantai sertifikat.
  - `--custom-private-key`- Kunci pribadi dalam format PEM untuk menghubungkan ke server dengan menggunakan HTTPS. Kunci privat tidak boleh dienkripsi; tidak dapat dilindungi oleh kata sandi atau frasa sandi.
- Pemeliharaan sistem mingguan diperlukan. Nilai yang valid harus ditentukan dalam format berikut:`DDD:HH:MM`. Waktu yang ditentukan adalah dalam Waktu Universal Terkoordinasi (UTC). Jika Anda tidak menentukan nilai untuk`--preferred-maintenance-window`, nilai default adalah acak, periode satu jam pada hari Selasa, Rabu, atau Jumat.
- Nilai yang valid untuk `--preferred-backup-window` harus ditentukan dalam salah satu format berikut: `HH:MM` untuk backup harian, atau `DDD:HH:MM` untuk backup mingguan. Waktu yang ditentukan adalah dalam UTC. Nilai default adalah waktu mulai acak harian. Untuk memilih keluar dari backup otomatis, tambahkan parameter `--disable-automated-backup` sebagai gantinya.
- Untuk`--security-group-ids`, masukkan satu atau lebih grup keamanan IDs, dipisahkan oleh spasi.
- Untuk`--subnet-ids`, masukkan subnet ID.

```
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
```

Contoh berikut membuat master Puppet yang menggunakan domain kustom.

```
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 untuk Puppet Enterprise membutuhkan waktu sekitar 15 menit untuk membuat server baru. Dalam output `create-server` perintah, salin nilai `Endpoint` atribut. Berikut adalah contohnya.

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

   Jangan mengabaikan output dari `create-server` perintah atau menutup sesi shell Anda, karena output dapat berisi informasi penting yang tidak ditampilkan lagi. Untuk mendapatkan kata sandi dan starter kit dari `create-server` hasil, lanjutkan ke langkah berikutnya.

5. Jika Anda memilih untuk membuat kata sandi OpsWorks untuk Puppet Enterprise untuk Anda, Anda dapat mengekstraknya dalam format yang dapat digunakan dari `create-server` hasil dengan menggunakan prosesor JSON seperti jq. Setelah Anda menginstal jq, Anda dapat

menjalankan perintah berikut untuk mengekstrak kata sandi administrator Puppet dan starter kit. Jika Anda tidak memberikan kata sandi Anda sendiri pada Langkah 3, pastikan untuk menyimpan kata sandi administrator yang diekstraksi di lokasi yang nyaman namun aman.

```
#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**
>
> Anda tidak dapat membuat ulang starter kit master Puppet baru di. AWS Management Console Saat Anda membuat master Puppet dengan menggunakan AWS CLI, jalankan `jq` perintah sebelumnya untuk menyimpan starter kit yang dikodekan base64 dalam hasil sebagai file ZIP. `create-server`

6. Secara opsional, jika Anda tidak mengekstrak starter kit dari hasil `create-server` perintah, Anda dapat mengunduh starter kit baru dari halaman Properties server di konsol OpsWorks for Puppet Enterprise.
7. Jika Anda tidak menggunakan domain khusus, lanjutkan ke langkah berikutnya. Jika Anda menggunakan domain kustom dengan server, buat entri CNAME di alat manajemen DNS perusahaan Anda untuk mengarahkan domain kustom Anda ke titik akhir OpsWorks untuk Puppet Enterprise yang Anda salin di langkah 4. Anda tidak dapat menjangkau atau masuk ke server dengan domain khusus sampai Anda menyelesaikan langkah ini.
8. Ketika proses pembuatan server selesai, lanjutkan keKonfigurasikan Puppet Master Menggunakan Starter Kit.

Lihat Juga

- Buat Master Perusahaan Boneka dengan menggunakan AWS CLI
- Cadangkan dan Kembalikan Server OpsWorks untuk Puppet Enterprise
- CreateServerdi Referensi AWS OpsWorks CM API
- create-serverdalam Referensi AWS CLI Perintah

Bekerja dengan Tag pada AWS OpsWorks for Puppet Enterprise Sumber Daya

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tag adalah kata atau frasa yang bertindak sebagai metadata untuk mengidentifikasi dan mengatur sumber daya AWS Anda. OpsWorks Untuk Puppet Enterprise, sumber daya dapat memiliki hingga 50 tag yang diterapkan pengguna. Setiap tag terdiri dari kunci dan satu nilai opsional. Anda dapat menerapkan tag ke sumber daya berikut di OpsWorks untuk Puppet Enterprise:

- OpsWorks untuk server Puppet Enterprise
- Cadangan OpsWorks untuk server Puppet Enterprise

Tag pada AWS sumber daya dapat membantu Anda melacak biaya, mengontrol akses ke sumber daya, mengelompokkan sumber daya untuk mengotomatisasi tugas, atau mengatur sumber daya berdasarkan tujuan atau tahap siklus hidup. Untuk informasi selengkapnya tentang manfaat tag, lihat Strategi Penandaan AWS di AWS Answers dan Menggunakan Tag Alokasi Biaya di AWS Manajemen Penagihan dan Biaya Panduan Pengguna.

Untuk menggunakan tag untuk mengontrol akses ke OpsWorks server atau cadangan Puppet Enterprise, buat atau edit pernyataan kebijakan di AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan AWS Identity and Access Management Pengguna.

Saat Anda menerapkan tag ke master OpsWorks for Puppet Enterprise, tag juga diterapkan ke cadangan master, bucket Amazon S3 yang menyimpan cadangan, instance EC2 Amazon master, rahasia untuk master yang disimpan AWS Secrets Manager, dan alamat IP Elastis yang digunakan oleh master. Tag tidak disebarkan ke AWS CloudFormation tumpukan yang AWS OpsWorks digunakan untuk membuat master boneka Anda.

Topik

- Bagaimana Tag Bekerja di AWS OpsWorks for Puppet Enterprise
- Menambahkan dan Mengelola Tag OpsWorks untuk Puppet Enterprise (Konsol)
- Menambahkan dan Mengelola Tag OpsWorks untuk Puppet Enterprise (CLI)
- Lihat Juga

Bagaimana Tag Bekerja di AWS OpsWorks for Puppet Enterprise

Dalam rilis ini, Anda dapat menambahkan dan mengelola tag dengan menggunakan AWS OpsWorks CM API atau file AWS Management Console. AWS OpsWorks CM juga mencoba menambahkan tag yang Anda tambahkan ke server ke AWS sumber daya yang terkait dengan server, termasuk EC2 instance, rahasia di Secrets Manager, alamat IP Elastis, grup keamanan, bucket S3, dan backup.

Tabel berikut memberikan ikhtisar tentang bagaimana Anda menambahkan dan mengelola tag OpsWorks untuk Puppet Enterprise.

Tindakan	Apa yang harus digunakan
Tambahkan tag ke server baru OpsWorks untuk Puppet Enterprise atau cadangan yang Anda buat secara manual.	• Pilih Buat server Perusahaan Boneka dan tambahkan tag pada halaman Konfigurasi pengaturan lanjutan. • Pilih Buat cadangan di halaman Backup untuk server yang ada, dan tambahkan tag pada halaman Buat cadangan dari server Puppet Enterprise Anda. • Tambahkan `Tags` parameter ke `CreateBackup` perintah `CreateServer` atau.
Lihat tag pada sumber daya.	• Pada halaman detail untuk server Anda, pilih Tag di panel navigasi. • Pada halaman Backup untuk server Anda, pilih cadangan, lalu pilih Edit cadangan. • Jalankan perintah `ListTagsForResource`.

Tindakan	Apa yang harus digunakan
Tambahkan tag ke server Puppet Enterprise atau cadangan yang sudah OpsWorks ada, terlepas dari apakah cadangan dibuat secara manual atau otomatis.	• Pada halaman detail untuk server Anda, pilih Tag di panel navigasi, lalu pilih Edit. • Pada halaman Backup untuk server Anda, pilih cadangan, lalu pilih Edit cadangan. • Jalankan perintah `TagResource` .
Hapus tag dari sumber daya.	• Pada halaman detail untuk server Anda, pilih Tag di panel navigasi, lalu pilih Edit. Pilih X di samping tag yang ingin Anda hapus. • Pada halaman Backup untuk server Anda, pilih cadangan, lalu pilih Edit cadangan. Pilih X di samping tag yang ingin Anda hapus. • Jalankan perintah `UntagResource` .

`DescribeServers`dan `DescribeBackups` tanggapan tidak termasuk informasi tag. Untuk menampilkan tag, gunakan `ListTagsForResource` API.

Menambahkan dan Mengelola Tag OpsWorks untuk Puppet Enterprise (Konsol)

Prosedur di bagian ini dilakukan di AWS Management Console.

Jika Anda menambahkan tag, kunci tag tidak bisa kosong. Kuncinya bisa maksimal 127 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: Nilai tag `+ - = . _ : / @` adalah opsional. Anda dapat menambahkan tag yang memiliki kunci, tetapi tidak ada nilai. Nilai dapat maksimal 255 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: `+ - = . _ : / @`

Topik

- Tambahkan Tag ke Server Perusahaan Boneka Baru OpsWorks (Konsol)
- Tambahkan Tag ke Cadangan Baru (Konsol)
- Menambahkan atau Melihat Tag di Server yang Ada (Konsol)
- Menambahkan atau Melihat Tag pada Cadangan yang Ada (Konsol)

- Hapus Tag dari Server (Konsol)
- Hapus Tag dari Backup (Konsol)

Tambahkan Tag ke Server Perusahaan Boneka Baru OpsWorks (Konsol)

1. Pastikan untuk menyelesaikan prasyarat apa pun untuk membuat master for Puppet Enterprise OpsWorks .
2. Ikuti langkah 1-8 diBuat Master Perusahaan Boneka dengan menggunakan AWS Management Console.
3. Setelah Anda menentukan pengaturan cadangan otomatis, tambahkan tag di area Tag pada halaman Konfigurasikan pengaturan lanjutan. Anda dapat menambahkan maksimal 50 tag. Setelah selesai menambahkan tag, pilih Berikutnya.
4. Lanjutkan ke langkah 11Buat Master Perusahaan Boneka dengan menggunakan AWS Management Console, dan tinjau pengaturan yang telah Anda pilih untuk server baru.

Tambahkan Tag ke Cadangan Baru (Konsol)

1. Pada halaman beranda OpsWorks for Puppet Enterprise, pilih Puppet master yang sudah ada.
2. Dari halaman detail server, pilih Backup di panel navigasi.
3. Pada halaman Backup, pilih Buat cadangan.
4. Tambahkan tag. Pilih Buat ketika Anda selesai menambahkan tag.

Menambahkan atau Melihat Tag di Server yang Ada (Konsol)

1. Pada halaman beranda OpsWorks for Puppet Enterprise, pilih master Boneka yang ada untuk membuka halaman detailnya.
2. Pilih Tag di panel navigasi, atau di bagian bawah halaman detail, pilih Lihat semua tag.
3. Pada halaman Tag, pilih Edit.
4. Tambahkan atau edit tag di server. Setelah selesai, pilih Simpan.

> **Note**
>
> Ketahuilah bahwa mengubah tag pada master Puppet Anda juga mengubah tag pada sumber daya yang terkait dengan server, seperti EC2 instance, alamat IP Elastis, grup keamanan, bucket S3, dan cadangan.

Menambahkan atau Melihat Tag pada Cadangan yang Ada (Konsol)

1. Pada halaman beranda OpsWorks for Puppet Enterprise, pilih master Boneka yang ada untuk membuka halaman detailnya.
2. Pilih Cadangan di panel navigasi, atau di area Pencadangan terbaru pada halaman detail, pilih Lihat semua cadangan.
3. Pada halaman Cadangan, pilih cadangan untuk dikelola, lalu pilih Edit cadangan.
4. Tambahkan atau edit tag pada cadangan. Pilih Perbarui saat Anda selesai.

Hapus Tag dari Server (Konsol)

1. Pada halaman beranda OpsWorks for Puppet Enterprise, pilih master Boneka yang ada untuk membuka halaman detailnya.
2. Pilih Tag di panel navigasi, atau di bagian bawah halaman detail, pilih Lihat semua tag.
3. Pada halaman Tag, pilih Edit.
4. Pilih X di samping tag untuk menghapus tag. Setelah selesai, pilih Simpan.

> **Note**
>
> Ketahuilah bahwa mengubah tag pada master Puppet Anda juga mengubah tag pada sumber daya yang terkait dengan server, seperti EC2 instance, alamat IP Elastis, grup keamanan, bucket S3, dan cadangan.

Hapus Tag dari Backup (Konsol)

1. Pada halaman beranda OpsWorks for Puppet Enterprise, pilih master Boneka yang ada untuk membuka halaman detailnya.

2. Pilih Cadangan di panel navigasi, atau di area Pencadangan terbaru pada halaman detail, pilih Lihat semua cadangan.
3. Pada halaman Cadangan, pilih cadangan untuk dikelola, lalu pilih Edit cadangan.
4. Pilih X di samping tag untuk menghapus tag. Pilih Perbarui saat Anda selesai.

Menambahkan dan Mengelola Tag OpsWorks untuk Puppet Enterprise (CLI)

Prosedur di bagian ini dilakukan di AWS CLI. Pastikan bahwa Anda menjalankan rilis terbaru AWS CLI sebelum Anda mulai bekerja dengan tag. Untuk informasi selengkapnya tentang menginstal atau memperbarui AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Jika Anda menambahkan tag, kunci tag tidak bisa kosong. Kuncinya bisa maksimal 127 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: Nilai tag `+ - = . _ : / @` adalah opsional. Anda dapat menambahkan tag yang memiliki kunci, tetapi tidak ada nilai. Nilai dapat maksimal 255 karakter, dan hanya berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: `+ - = . _ : / @`

Topik

- Tambahkan Tag ke Server Perusahaan Boneka Baru OpsWorks (CLI)
- Tambahkan Tag ke Cadangan Baru (CLI)
- Tambahkan Tag ke Server atau Cadangan yang Ada (CLI)
- Daftar Sumber Daya Tag (CLI)
- Hapus Tag dari Sumber Daya (CLI)

Tambahkan Tag ke Server Perusahaan Boneka Baru OpsWorks (CLI)

Anda dapat menggunakan AWS CLI untuk menambahkan tag saat Anda membuat server OpsWorks for Puppet Enterprise. Prosedur ini tidak menjelaskan secara lengkap cara membuat server. Untuk informasi rinci tentang cara membuat server OpsWorks untuk Puppet Enterprise dengan menggunakan AWS CLI, lihat Buat Master Perusahaan Boneka dengan menggunakan AWS CLI di panduan ini. Anda dapat menambahkan hingga 50 tag ke server.

1. Pastikan untuk menyelesaikan prasyarat apa pun untuk membuat server for Puppet Enterprise OpsWorks .

2. Selesaikan langkah 1-4 dari. Buat Master Perusahaan Boneka dengan menggunakan AWS CLI
3. Untuk langkah 5, saat Anda menjalankan `create-server` perintah, tambahkan `--tags` parameter ke perintah, seperti yang ditunjukkan pada contoh berikut.

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

Berikut ini adalah contoh yang hanya menampilkan bagian tag dari `create-server` perintah.

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

4. Selesaikan langkah-langkah selebihnya di Buat Master Perusahaan Boneka dengan menggunakan AWS CLI. Untuk memverifikasi bahwa tag Anda telah ditambahkan ke server baru, ikuti langkah-langkah Daftar Sumber Daya Tag (CLI) dalam topik ini.

Tambahkan Tag ke Cadangan Baru (CLI)

Anda dapat menggunakan AWS CLI untuk menambahkan tag saat Anda membuat cadangan manual baru dari server OpsWorks for Puppet Enterprise. Prosedur ini tidak menjelaskan secara lengkap cara membuat cadangan manual. Untuk informasi terperinci tentang cara membuat cadangan manual, lihat “Untuk melakukan pencadangan manual di AWS CLI" diCadangkan OpsWorks untuk Server Perusahaan Boneka. Anda dapat menambahkan hingga 50 tag ke cadangan. Jika server memiliki tag, backup baru secara otomatis ditandai dengan tag server.

Secara default, saat Anda membuat server baru OpsWorks untuk Puppet Enterprise, pencadangan otomatis diaktifkan. Anda dapat menambahkan tag ke cadangan otomatis dengan menjalankan `tag-resource` perintah, yang dijelaskan Tambahkan Tag ke Server atau Cadangan yang Ada (CLI) dalam topik ini.

- Untuk menambahkan tag ke cadangan manual saat Anda membuat cadangan, jalankan perintah berikut. Hanya bagian tag dari perintah yang ditampilkan. Untuk contoh `create-backup` perintah lengkap, lihat “Untuk melakukan pencadangan manual di AWS CLI" inCadangkan OpsWorks untuk Server Perusahaan Boneka.

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

Contoh berikut hanya menunjukkan bagian tag dari `create-backup` perintah.

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

Tambahkan Tag ke Server atau Cadangan yang Ada (CLI)

Anda dapat menjalankan `tag-resource` perintah untuk menambahkan tag yang ada OpsWorks untuk server atau cadangan Puppet Enterprise (apakah cadangan dibuat secara otomatis atau manual). Tentukan Nomor Sumber Daya Amazon (ARN) dari sumber daya target untuk menambahkan tag ke dalamnya.

1. Untuk mendapatkan ARN dari sumber daya yang ingin Anda terapkan tag:

   - Untuk server, jalankan`describe-servers --server-name` *`server_name`*. Hasil perintah menunjukkan server ARN.
   - Untuk cadangan, jalankan`describe-backups --backup-id` *`backup_ID`*. Hasil perintah menunjukkan ARN cadangan. Anda juga dapat menjalankan `describe-backups --server-name` *`server_name`* untuk menampilkan informasi tentang semua cadangan untuk server spesifik OpsWorks untuk Puppet Enterprise.

   Contoh berikut hanya menunjukkan hasil `describe-servers --server-name opsworks-cm-test` perintah. `ServerArn` `ServerArn`Nilai ditambahkan ke `tag-resource` perintah untuk menambahkan tag ke server.

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

2. Jalankan `tag-resource` perintah dengan ARN yang Anda kembalikan pada langkah 1.

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

Berikut adalah contohnya.

```
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. Untuk memverifikasi bahwa tag berhasil ditambahkan, lanjutkan ke prosedur berikutnya,Daftar Sumber Daya Tag (CLI).

Daftar Sumber Daya Tag (CLI)

Anda dapat menjalankan `list-tags-for-resource` perintah untuk menampilkan tag yang dilampirkan OpsWorks untuk server atau cadangan Puppet Enterprise. Tentukan ARN dari sumber daya target untuk melihat tag-nya.

1. Untuk mendapatkan ARN dari sumber daya yang ingin Anda cantumkan tag:
   - Untuk server, jalankan`describe-servers --server-name` *`server_name`*. Hasil perintah menunjukkan server ARN.
   - Untuk cadangan, jalankan`describe-backups --backup-id` *`backup_ID`*. Hasil perintah menunjukkan ARN cadangan. Anda juga dapat menjalankan `describe-backups --server-name` *`server_name`* untuk menampilkan informasi tentang semua cadangan untuk server spesifik OpsWorks untuk Puppet Enterprise.
2. Jalankan `list-tags-for-resource` perintah dengan ARN yang Anda kembalikan pada langkah 1.

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

Berikut adalah contohnya.

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

Jika ada tag pada sumber daya, perintah mengembalikan hasil seperti berikut ini.

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

Hapus Tag dari Sumber Daya (CLI)

Anda dapat menjalankan `untag-resource` perintah untuk menghapus tag dari OpsWorks server atau cadangan Puppet Enterprise. Jika sumber daya dihapus, tag pada sumber daya juga dihapus. Tentukan Nomor Sumber Daya Amazon (ARN) dari sumber daya target untuk menghapus tag darinya.

1. Untuk mendapatkan ARN dari sumber daya dari mana Anda ingin menghapus tag:

   - Untuk server, jalankan`describe-servers --server-name` *`server_name`*. Hasil perintah menunjukkan server ARN.
   - Untuk cadangan, jalankan`describe-backups --backup-id` *`backup_ID`*. Hasil perintah menunjukkan ARN cadangan. Anda juga dapat menjalankan `describe-backups --server-name` *`server_name`* untuk menampilkan informasi tentang semua cadangan untuk server spesifik OpsWorks untuk Puppet Enterprise.

2. Jalankan `untag-resource` perintah dengan ARN yang Anda kembalikan pada langkah 1. Tentukan hanya tag yang ingin Anda hapus.

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

   Dalam contoh ini, `untag-resource` perintah hanya menghapus tag dengan kunci dari `Stage` dan nilai`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. Untuk memverifikasi bahwa tag berhasil dihapus, ikuti langkah-langkah Daftar Sumber Daya Tag (CLI) dalam topik ini.

Lihat Juga

- Buat Master Perusahaan Boneka dengan menggunakan AWS CLI
- Cadangkan OpsWorks untuk Server Perusahaan Boneka
- Strategi Penandaan AWS
- Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan AWS Identity and Access Management Pengguna
- Menggunakan Tanda Alokasi Biaya dalam AWS Manajemen Penagihan dan Biaya Panduan Pengguna
- CreateBackupdi Referensi AWS OpsWorks CM API
- CreateServerdi Referensi AWS OpsWorks CM API
- TagResourcedi Referensi AWS OpsWorks CM API
- ListTagsForResourcedi Referensi AWS OpsWorks CM API
- UntagResourcedi Referensi AWS OpsWorks CM API

Cadangkan dan Kembalikan Server OpsWorks untuk Puppet Enterprise

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara mencadangkan dan memulihkan server OpsWorks untuk Perusahaan Boneka.

Topik

- Cadangkan OpsWorks untuk Server Perusahaan Boneka
- Mengembalikan OpsWorks untuk Puppet Enterprise Server dari Backup

Cadangkan OpsWorks untuk Server Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menentukan pengulangan harian atau mingguan OpsWorks untuk cadangan server Puppet Enterprise, dan meminta layanan menyimpan cadangan di Amazon Simple Storage Service (Amazon S3) atas nama Anda. Atau, Anda dapat membuat cadangan manual sesuai permintaan.

Karena cadangan disimpan di Amazon S3, cadangan tersebut dikenakan biaya tambahan. Anda dapat menentukan periode retensi cadangan hingga 30 generasi. Anda dapat mengirimkan permintaan layanan agar batas tersebut diubah dengan menggunakan saluran AWS dukungan. Konten yang dikirim ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Cara Menghapus Bucket S3? .

Anda dapat menambahkan tag ke cadangan master OpsWorks for Puppet Enterprise. Jika Anda telah menambahkan tag ke master OpsWorks for Puppet Enterprise, backup otomatis master Puppet mewarisi tag tersebut. Untuk informasi selengkapnya tentang cara menambahkan dan mengelola tag pada cadangan, lihat Bekerja dengan Tag pada AWS OpsWorks for Puppet Enterprise Sumber Daya di panduan ini.

Topik

- Pencadangan Otomatis

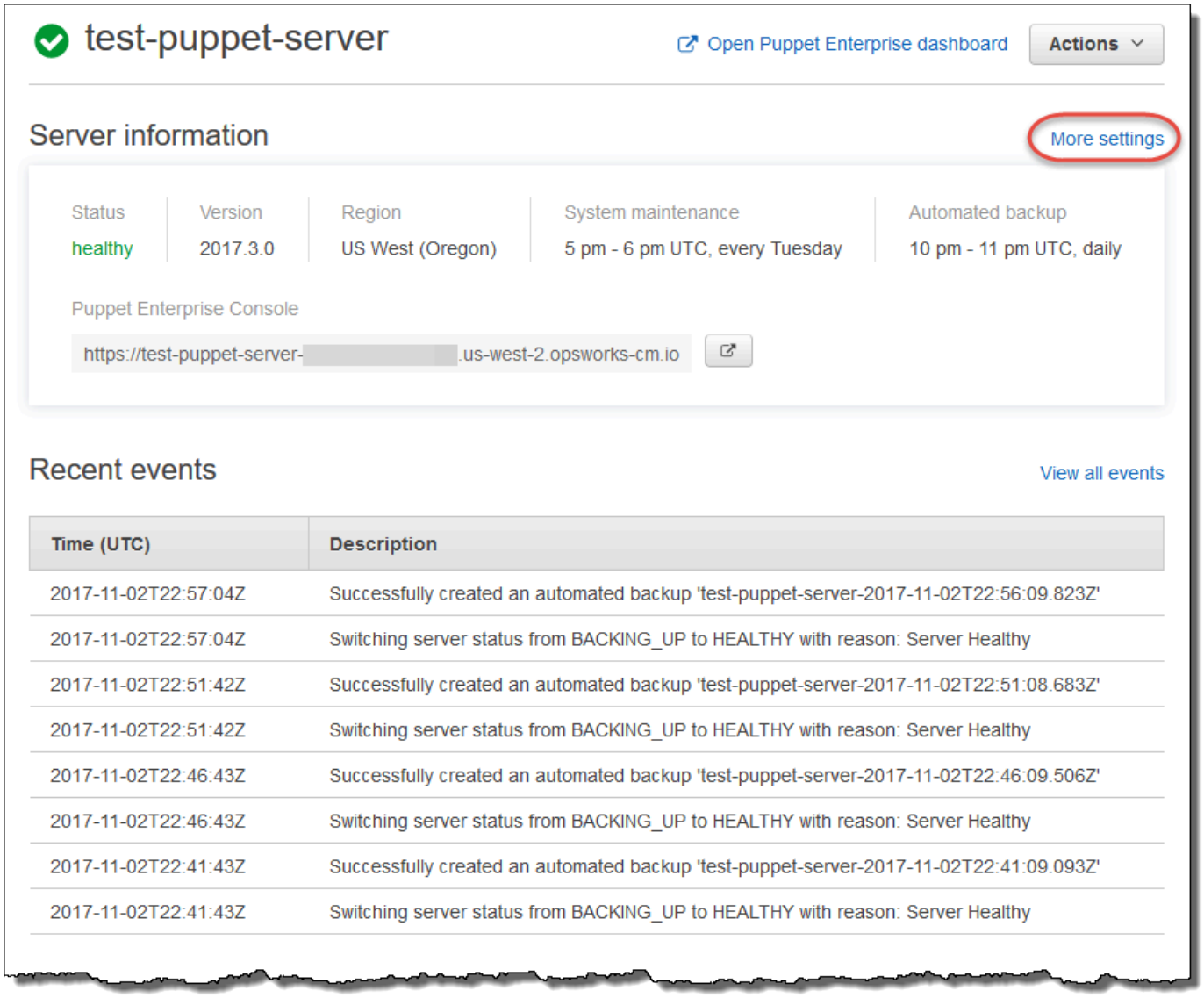
- Backup Manual
- Hapus cadangan

Pencadangan Otomatis

Saat Anda mengonfigurasi server OpsWorks untuk Puppet Enterprise, Anda memilih pencadangan otomatis atau manual. OpsWorks untuk Puppet Enterprise memulai pencadangan otomatis selama satu jam dan pada hari yang Anda pilih di bagian Pencadangan otomatis pada halaman Konfigurasi pengaturan lanjutan dari wizard penyiapan. Setelah server Anda online, Anda dapat mengubah pengaturan cadangan dengan melakukan langkah-langkah berikut pada halaman properti server.

Untuk mengubah pengaturan cadangan otomatis

1. Pada halaman properti server, pilih Pengaturan lainnya.



2. Untuk mematikan pencadangan otomatis, pilih Tidak untuk opsi Aktifkan pencadangan otomatis. Simpan perubahan Anda; Anda tidak perlu melanjutkan ke langkah berikutnya.
3. Di bagian Backup Otomatis, ubah frekuensi, waktu mulai, atau generasi yang akan disimpan. Simpan perubahan Anda.

Backup Manual

Anda dapat memulai pencadangan manual kapan saja di AWS Management Console, atau dengan menjalankan perintah AWS CLI create-backup. Pencadangan manual tidak termasuk dalam maksimum 30 generasi cadangan otomatis yang disimpan. Maksimal 10 cadangan manual disimpan, dan harus dihapus secara manual dari Amazon S3.

Untuk melakukan backup manual di AWS Management Console

1. Pada halaman server Puppet Enterprise, pilih server yang ingin Anda backup.
2. Pada halaman properti untuk server, di sebelah kiri panel navigasi, pilih cadangan.
3. Pilih Buat cadangan.
4. Pencadangan manual selesai ketika halaman menunjukkan tanda centang hijau di kolom Status cadangan.

Untuk melakukan backup manual di AWS CLI

Anda dapat menambahkan tag ketika Anda membuat cadangan manual baru dari server OpsWorks for Puppet Enterprise. Untuk informasi selengkapnya tentang cara menambahkan tag saat Anda membuat cadangan manual, lihatTambahkan Tag ke Cadangan Baru (CLI).

- Untuk memulai cadangan manual, jalankan AWS CLI perintah berikut.

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

Hapus cadangan

Menghapus cadangan secara permanen menghapusnya dari bucket S3 tempat cadangan disimpan.

Untuk menghapus cadangan di AWS Management Console

1. Pada halaman server Puppet Enterprise, pilih server yang ingin Anda backup.
2. Pada halaman properti untuk server, di sebelah kiri panel navigasi, pilih cadangan.
3. Pilih cadangan yang ingin Anda hapus, dan pilih Hapus cadangan. Anda hanya dapat memilih satu cadangan dalam satu waktu.
4. Ketika Anda diminta untuk mengonfirmasi penghapusan, isi kotak centang untuk Hapus cadangan, yang disimpan dalam ember S3, lalu pilih Ya, Hapus.

Untuk menghapus cadangan di AWS CLI

- Untuk menghapus cadangan, jalankan AWS CLI perintah berikut, ganti nilai `--backup-id` dengan ID cadangan yang ingin Anda hapus. Backup IDs dalam format*ServerName-yyyyMMddHHmmssSSS*. Misalnya, **`puppet-server-20171218132604388`**.

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

Mengembalikan OpsWorks untuk Puppet Enterprise Server dari Backup

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah menelusuri cadangan yang tersedia, Anda dapat dengan mudah memilih titik waktu untuk memulihkan server Puppet OpsWorks Enterprise Anda. Pencadangan server berisi data persisten perangkat lunak manajemen konfigurasi seperti modul, kelas, asosiasi simpul, informasi basis data (termasuk laporan, fakta, dll.). Melakukan restorasi server di tempat (yaitu, memulihkan yang ada OpsWorks untuk server Puppet Enterprise ke EC2 instance baru) mendaftarkan ulang node yang terdaftar pada saat pencadangan yang Anda gunakan untuk memulihkan server, dan mengalihkan lalu lintas ke instance baru jika restorasi berhasil, dan status server Puppet Enterprise dipulihkan OpsWorks . `Healthy` Memulihkan ke server Puppet Enterprise yang baru dibuat OpsWorks tidak mempertahankan koneksi node. Memulihkan server tidak memperbarui versi perangkat lunak Puppet; itu menerapkan versi Puppet yang sama dan data manajemen konfigurasi yang tersedia di cadangan yang Anda pilih.

Memulihkan server biasanya membutuhkan lebih banyak waktu daripada membuat server baru; waktu tergantung pada ukuran cadangan yang Anda pilih. Setelah restorasi selesai, EC2 contoh lama tetap dalam `Stopped` keadaan `Running` atau, tetapi hanya sementara. Akhirnya dihentikan.

Dalam rilis ini, Anda dapat menggunakan AWS CLI untuk mengembalikan master Boneka OpsWorks untuk Perusahaan Boneka.

> **Note**
>
> Anda juga dapat menjalankan perintah restore-server untuk mengubah jenis instans saat ini, atau untuk memulihkan atau mengatur kunci SSH Anda jika hilang atau disusupi.

Untuk memulihkan server dari cadangan

1. Dalam AWS CLI, jalankan perintah berikut untuk mengembalikan daftar backup yang tersedia dan mereka. IDs Buatlah sebuah catatan ID cadangan yang ingin Anda gunakan. Backup IDs dalam format*myServerName-yyyyMMddHHmmssSSS*.

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

2. Jalankan perintah berikut.

   ```
   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"
   ```

   Berikut adalah contohnya.

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

3. Tunggu sampai restorasi selesai.

Pemeliharaan Sistem OpsWorks untuk Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pemeliharaan sistem wajib memastikan bahwa versi terbaru AWS dari Puppet Server, termasuk pembaruan keamanan, selalu berjalan di server OpsWorks for Puppet Enterprise. Pemeliharaan sistem diperlukan minimal seminggu sekali. Dengan menggunakan AWS CLI, Anda dapat mengkonfigurasi pemeliharaan otomatis harian, jika diinginkan. Anda juga dapat menggunakan AWS CLI untuk melakukan pemeliharaan sistem sesuai permintaan, selain pemeliharaan sistem terjadwal.

Ketika versi baru perangkat lunak Puppet tersedia, pemeliharaan sistem dirancang untuk memperbarui versi Puppet Server di server secara otomatis, segera setelah melewati pengujian AWS. AWS melakukan pengujian ekstensif untuk memverifikasi bahwa upgrade Puppet siap produksi dan tidak mengganggu lingkungan pelanggan yang ada, sehingga mungkin ada kelambatan antara rilis perangkat lunak Puppet dan ketersediaannya untuk aplikasi yang ada untuk server Puppet Enterprise. OpsWorks Untuk memperbarui versi perangkat lunak Boneka yang tersedia sesuai permintaan, lihat Memulai pemeliharaan sistem sesuai permintaan di topik ini.

Pemeliharaan sistem meluncurkan instans baru dari cadangan yang dilakukan sebagai bagian dari proses pemeliharaan, yang membantu mengurangi risiko dari EC2 instans Amazon yang terdegradasi atau terganggu yang menjalani pemeliharaan berkala.

> **Important**
>
> Pemeliharaan sistem menghapus file atau konfigurasi khusus yang telah Anda tambahkan ke server OpsWorks for Puppet Enterprise. Untuk informasi selengkapnya tentang cara memperbaiki konfigurasi atau kehilangan file, lihat Memulihkan konfigurasi dan file khusus setelah pemeliharaan di topik ini.

Topik

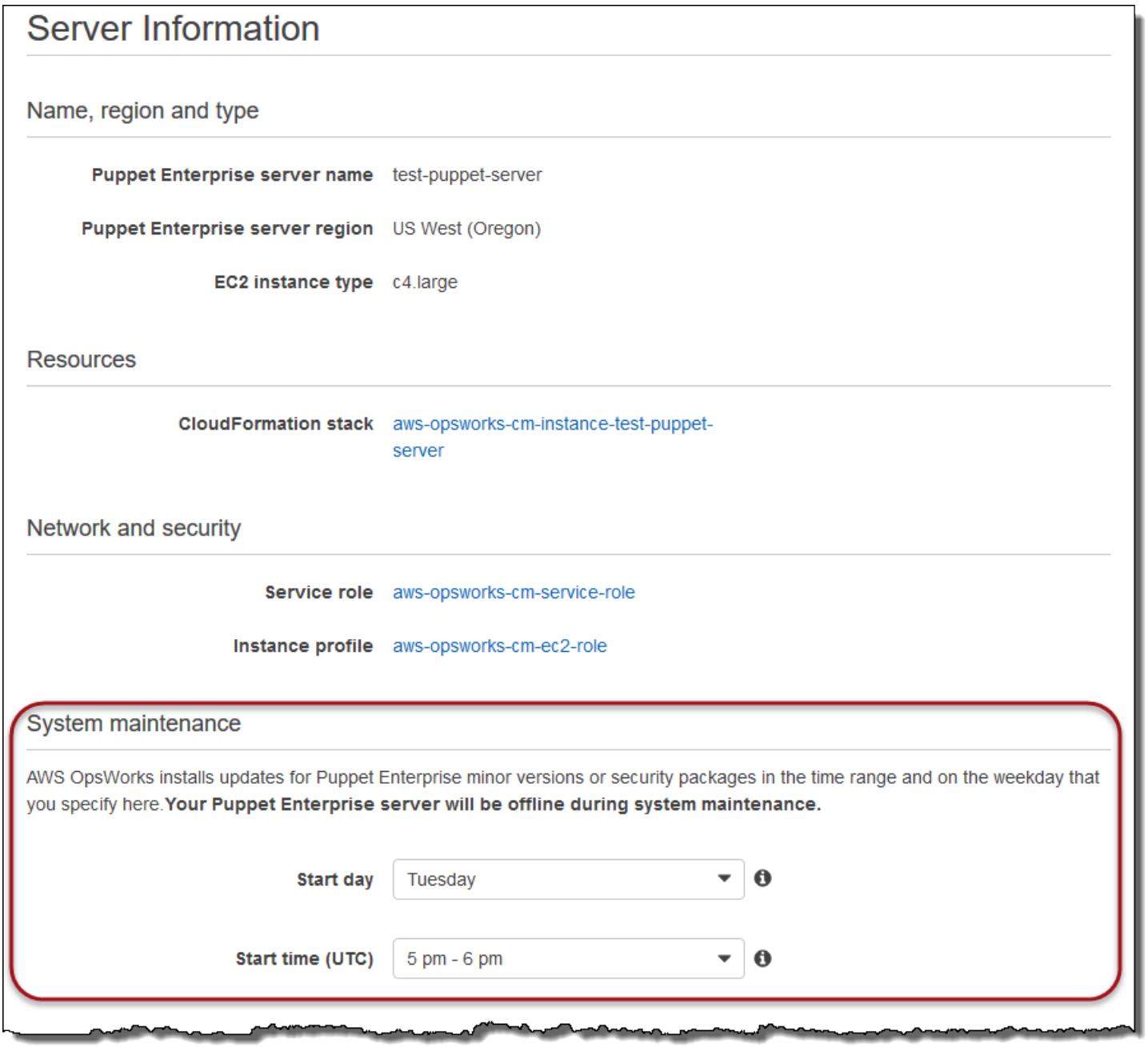
- Mengkonfigurasi pemeliharaan sistem
- Memulai pemeliharaan sistem sesuai permintaan
- Memulihkan konfigurasi dan file khusus setelah pemeliharaan

Mengkonfigurasi pemeliharaan sistem

Saat Anda membuat server baru OpsWorks untuk Puppet Enterprise, Anda dapat mengonfigurasi hari kerja dan waktu, di Coordinated Universal Time (UTC), agar pemeliharaan sistem dimulai. Pemeliharaan dimulai selama jam yang Anda tentukan. Karena Anda harus mengharapkan server

offline selama pemeliharaan sistem, pilih waktu permintaan server yang rendah dalam jam kantor biasa. Status server `UNDER_MAINTENANCE` saat pemeliharaan sedang berlangsung.

Anda juga dapat mengubah pengaturan pemeliharaan sistem pada server Puppet Enterprise yang ada OpsWorks , dengan mengubah pengaturan di area pemeliharaan sistem halaman Pengaturan untuk server Anda, seperti yang ditunjukkan pada gambar berikut.



Di bagian Pemeliharaan sistem, atur hari dan jam yang Anda inginkan untuk memulai pemeliharaan sistem.

Mengkonfigurasi pemeliharaan sistem dengan menggunakan AWS CLI

Anda juga dapat mengonfigurasi waktu mulai otomatis pemeliharaan sistem dengan menggunakan AWS CLI. Ini AWS CLI memungkinkan Anda mengonfigurasi pemeliharaan otomatis harian, jika diinginkan, dengan menghilangkan awalan hari kerja tiga karakter.

Dalam sebuah `create-server` perintah, tambahkan `--preferred-maintenance-window` parameter ke perintah Anda, setelah menentukan persyaratan untuk membuat instance server (seperti jenis instance, ARN profil instance, dan ARN peran layanan). Dalam `create-server` contoh berikut, `--preferred-maintenance-window` diatur ke`Mon:08:00`, artinya Anda telah mengatur pemeliharaan untuk dimulai setiap Senin pagi pukul 8:00 pagi. 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"
```

Dalam sebuah `update-server` perintah, Anda dapat memperbarui `--preferred-maintenance-window` nilainya sendiri, jika diinginkan. Dalam contoh berikut, jendela pemeliharaan diatur ke Jumat malam pukul 18:15. UTC.

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

Untuk mengubah waktu mulai jendela pemeliharaan menjadi 6:15 p.m. UTC setiap hari, hilangkan awalan tiga karakter hari kerja, seperti yang ditunjukkan pada contoh berikut.

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

Untuk informasi selengkapnya tentang pengaturan jendela pemeliharaan sistem pilihan dengan menggunakan AWS CLI, lihat create-server dan update-server.

Memulai pemeliharaan sistem sesuai permintaan

Untuk memulai pemeliharaan sistem sesuai permintaan, di luar pemeliharaan otomatis mingguan atau harian yang dikonfigurasi, jalankan AWS CLI perintah berikut. Anda tidak dapat memulai pemeliharaan sesuai permintaan di. AWS Management Console

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

Untuk informasi selengkapnya tentang perintah ini, lihat pemeliharaan awal.

Memulihkan konfigurasi dan file khusus setelah pemeliharaan

Pemeliharaan sistem dapat menghapus atau mengubah file atau konfigurasi khusus yang telah Anda tambahkan ke server Puppet Enterprise Anda OpsWorks .

Jika, setelah pemeliharaan dijalankan, master Puppet Anda kehilangan file atau pengaturan yang Anda tambahkan dengan menggunakan `RunCommand` atau SSH, Anda dapat menggunakan Amazon Machine Image (AMI) untuk meluncurkan instans Amazon EC2 baru. AMIs tersedia yang dibangun dari konfigurasi pra-pemeliharaan server.

Instance baru dalam keadaan yang sama dengan master Puppet sebelum pemeliharaan, dan harus menyertakan file dan pengaturan Anda yang hilang.

> **Important**
>
> Anda tidak dapat menggunakan instance baru untuk memulihkan server Anda; instance tidak dapat dijalankan sebagai master Puppet. Anda dapat menggunakan instance hanya untuk memulihkan file dan pengaturan konfigurasi Anda.

Untuk meluncurkan EC2 instance dari AMI, di EC2 konsol Amazon, buka panduan Peluncuran, pilih My AMIs, lalu pilih AMI yang memiliki nama server Anda. Ikuti langkah-langkah EC2 wizard Amazon seperti yang Anda lakukan untuk peluncuran instans lainnya.

Menambahkan Node Secara Otomatis OpsWorks untuk Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera

mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini menjelaskan cara menambahkan node Amazon Elastic Compute Cloud (Amazon EC2) ke server OpsWorks for Puppet Enterprise Anda secara otomatis. DiTambahkan Node untuk Dikelola oleh Puppet Master, Anda belajar cara menggunakan `associate-node` perintah untuk menambahkan satu node pada satu waktu ke server Puppet Enterprise Anda. Kode dalam topik ini menunjukkan cara menambahkan node secara otomatis menggunakan metode tanpa pengawasan. Metode yang direkomendasikan untuk asosiasi node baru tanpa pengawasan (atau otomatis) adalah dengan mengonfigurasi data EC2 pengguna Amazon. Secara default, server OpsWorks untuk Puppet Enterprise sudah `puppet-agent`tersedia untuk sistem operasi node Ubuntu, Amazon Linux, dan RHEL.

Untuk informasi tentang cara memisahkan node, lihat Putuskan Node dari Server OpsWorks for Puppet Enterprise di panduan ini, dan `disassociate-node`di dokumentasi OpsWorks for Puppet Enterprise API.

Langkah 1: Buat Peran IAM untuk Digunakan sebagai Profil Instance Anda

Buat peran AWS Identity and Access Management (IAM) untuk digunakan sebagai profil EC2 instans Anda, dan lampirkan kebijakan berikut ke peran IAM. Kebijakan ini memungkinkan `opsworks-cm` API untuk berkomunikasi dengan EC2 instance selama pendaftaran node. Untuk informasi selengkapnya tentang profil instans, lihat Menggunakan Profil Instance di EC2 dokumentasi Amazon. Untuk informasi tentang cara membuat peran IAM, lihat Membuat Peran IAM di Konsol di dokumentasi Amazon EC2 .

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

```
        }
    ]
}
```

Langkah 2: Buat Instans dengan Menggunakan Skrip Asosiasi Tanpa Pengawasan

Untuk membuat EC2 instance, Anda dapat menyalin skrip data pengguna yang disertakan dalam Starter Kit ke `userdata` bagian instruksi EC2 instance, konfigurasi peluncuran grup Amazon EC2 Auto Scaling, atau templat. AWS CloudFormation Skrip ini didukung hanya untuk EC2 instance yang menjalankan sistem operasi Ubuntu dan Amazon Linux. Untuk informasi selengkapnya tentang menambahkan skrip ke data pengguna, lihat Menjalankan Perintah pada Instans Linux Anda saat Peluncuran di EC2 dokumentasi Amazon. Cara termudah untuk membuat node baru adalah dengan menggunakan wizard peluncuran EC2 instans Amazon. Panduan ini menggunakan pengaturan modul contoh server web Apache yang dijelaskan dalam. Memulai dengan OpsWorks untuk Perusahaan Boneka

1. Skrip data pengguna di Starter Kit menjalankan `associate-node`perintah `opsworks-cm` API untuk mengaitkan node baru dengan master Puppet Anda. Dalam rilis ini, ia juga menginstal versi saat ini dari AWS CLI pada node untuk Anda, jika itu belum menjalankan up-to-date versi terbanyak. Simpan skrip ini ke lokasi yang nyaman sebagai`userdata.sh`.

   Secara default, nama node terdaftar baru adalah ID instance.
2. Ikuti prosedur dalam Meluncurkan Instance dalam EC2 dokumentasi, dengan modifikasi di sini. Di wizard peluncuran EC2 instance, pilih AMI Amazon Linux.
3. Pada halaman Konfigurasi Detail Instance, pilih MyPuppetInstanceProfile, peran yang Anda buat, sebagai peran IAM Anda. Langkah 1: Buat Peran IAM untuk Digunakan sebagai Profil Instance Anda
4. Di area Detail Lanjutan, unggah `userdata.sh` skrip yang Anda buat di Langkah 1.
5. Tidak diperlukan perubahan pada halaman Add Storage. Lanjutkan ke Tambahkan Tag.

   Dengan menerapkan tag ke EC2 instans Anda, Anda dapat menyesuaikan perilaku`userdata.sh`. Untuk contoh ini, terapkan peran `apache_webserver` ke node Anda dengan menambahkan tag berikut:**`pp_role`**, dengan nilainya**`apache_webserver`**.

   Menyetel `pp_role` nilai pada node menetapkan nilai data yang disimpan secara permanen dalam sertifikat agen node, memungkinkan klasifikasi node yang tepercaya. Untuk informasi

selengkapnya, lihat Permintaan ekstensi (data sertifikat permanen) dalam dokumentasi platform Wayang.

6. Pada halaman Configure Security Group, pilih Add Rule, dan kemudian pilih tipe HTTP untuk membuka port 8080 untuk server web Apache dalam contoh ini.
7. Pilih Tinjau dan Luncurkan, lalu pilih Luncurkan. Ketika node baru Anda dimulai, itu menerapkan konfigurasi Apache dari modul sampel yang Anda atur. Mengatur Contoh Starter Kit Apache
8. Ketika Anda membuka halaman web yang ditautkan ke DNS publik dari node baru Anda, Anda akan melihat situs web yang di-host oleh server web Apache yang dikelola Puppet Anda.

Putuskan Node dari Server OpsWorks for Puppet Enterprise

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara memisahkan, atau menghapus, node terkelola dari manajemen oleh server OpsWorks for Puppet Enterprise. Operasi ini dilakukan pada baris perintah atau di konsol Puppet Enterprise; Anda tidak dapat memisahkan node di konsol manajemen OpsWorks for Puppet Enterprise. Saat ini, OpsWorks for Puppet Enterprise API tidak mengizinkan penghapusan batch beberapa node. Perintah di bagian ini memisahkan satu node pada satu waktu.

Kami menyarankan Anda memisahkan node dari master Puppet sebelum Anda menghapus server, sehingga node terus beroperasi tanpa mencoba untuk terhubung kembali dengan server. Untuk melakukan ini, jalankan `disassociate-node` AWS CLI perintah. Untuk benar-benar menghapus node dari PE, Anda harus memisahkan node dan mencabut sertifikatnya, sehingga node tidak terus mencoba untuk memeriksa dengan master Puppet. Anda juga harus menghapus instalasi `puppet-agent` dari node ketika Anda tidak lagi ingin mengelolanya dengan menggunakan master Puppet.

Untuk memisahkan node

1. Dalam AWS CLI, jalankan perintah berikut untuk memisahkan node. *`Node_name`*adalah nama node yang ingin Anda lepaskan; untuk EC2 instance Amazon, ini adalah ID instance.

*Server_name*adalah nama master boneka dari mana Anda ingin memisahkan node. Kedua parameter diperlukan. `--region`Parameter tidak diperlukan kecuali Anda ingin memisahkan node dari master Puppet yang tidak ada di wilayah default Anda.

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

Berikut adalah contoh perintah tersebut.

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

2. Tunggu sampai pesan respons menunjukkan bahwa disasosiasi selesai.

Untuk informasi selengkapnya tentang cara menghapus server OpsWorks for Puppet Enterprise, lihatHapus OpsWorks untuk Puppet Enterprise Server.

Lihat Juga

- Hapus node dalam dokumentasi Puppet Enterprise

Hapus OpsWorks untuk Puppet Enterprise Server

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara menghapus server OpsWorks untuk Puppet Enterprise. Menghapus server juga menghapus peristiwa, log, dan modul apa pun yang disimpan di server. Sumber daya pendukung (instans Amazon Elastic Compute Cloud, volume Amazon Elastic Block Store, dll.) juga dihapus, bersama dengan semua pencadangan otomatis.

Meskipun menghapus server tidak menghapus node, mereka tidak lagi dikelola oleh server yang dihapus, dan akan terus mencoba untuk menyambung kembali. Untuk alasan ini, kami sarankan untuk memisahkan node terkelola sebelum Anda menghapus master Puppet. Dalam rilis ini, Anda dapat memisahkan node dengan menjalankan AWS CLI perintah.

Langkah 1: Pisahkan Node yang Dikelola

Putuskan node dari master Puppet sebelum Anda menghapus server, sehingga node terus beroperasi tanpa mencoba untuk terhubung kembali dengan server. Untuk melakukan ini, jalankan `disassociate-node` AWS CLI perintah.

Untuk memisahkan node

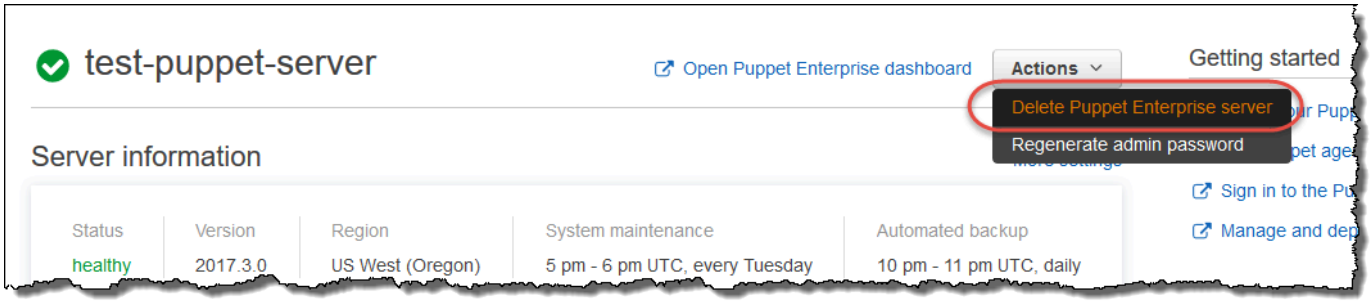
1. Dalam AWS CLI, jalankan perintah berikut untuk memisahkan node. *`Server_name`*adalah nama master boneka dari mana Anda ingin memisahkan node. Nilai `--node-name` dapat berupa ID instance.

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

2. Tunggu sampai pesan respons menunjukkan bahwa disasosiasi selesai.

Langkah 2: Hapus Server

1. Pada ubin server di dasbor, perluas menu Tindakan.



2. Pilih Hapus server Perusahaan Boneka.
3. Saat Anda diminta untuk mengonfirmasi penghapusan, isi kotak centang untuk menghapus peran dan sumber daya terkait, lalu pilih Ya, Hapus.

Lihat Juga

- Putuskan Node dari Server OpsWorks for Puppet Enterprise

Cara memigrasikan server OpsWorks for Puppet Enterprise ke Amazon Elastic Compute Cloud (Amazon) EC2

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Petunjuk di bawah ini menjelaskan cara memigrasi server Puppet Enterprise yang ada ke Amazon EC2, jika Anda ingin terus menggunakan Puppet Enterprise untuk kebutuhan manajemen konfigurasi Anda di luar. OpsWorks

Topik

- Langkah 1: Hubungi Puppet untuk membeli lisensi
- Langkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks
- Langkah 3: Buat cadangan server Puppet Enterprise Anda OpsWorks
- Langkah 4: Luncurkan EC2 instance baru
- Langkah 5: Instal Puppet Enterprise pada instance baru EC2
- Langkah 6: Kembalikan cadangan pada EC2 instance baru
- Langkah 7: Konfigurasikan lisensi Boneka Anda
- Langkah 8: Migrasikan node Anda
- Langkah 9: Hapus server Puppet Enterprise Anda OpsWorks

Langkah 1: Hubungi Puppet untuk membeli lisensi

Ketika Anda memigrasikan server Anda ke EC2, instance baru tidak datang dengan lisensi Puppet. Untuk membeli kunci lisensi, ikuti instruksi di situs web Boneka.

Langkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks

Temukan dan simpan nilai untuk server OpsWorks for Puppet Enterprise Anda.

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   Salin nama bucket Amazon S3 yang ada untuk server for Puppet OpsWorks Enterprise Anda. Nama bucket memiliki format: `aws-opsworks-cm-`*`server-name`*`-`*`random-string`*

2. Jalankan aws opsworks-cm describe-servers perintah untuk mendapatkan konfigurasi untuk server Puppet Enterprise Anda OpsWorks .

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

   Simpan nilai untuk`InstanceType`,`KeyPair`,`SubnetIds`,`SecurityGroupIds`,`InstanceProfileArn`, dan `Endpoint` dari respons.

3. Gunakan SSH untuk terhubung ke server Puppet Enterprise yang ada OpsWorks . Anda dapat menggunakan Session Manager di EC2 konsol alih-alih SSH.

   Jalankan perintah berikut.

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

   Respons menyediakan versi Puppet Enterprise (misalnya, 2019.8.10). Simpan nilai ini.

Anda akan menggunakan SSH atau Session manager untuk langkah selanjutnya.

Langkah 3: Buat cadangan server Puppet Enterprise Anda OpsWorks

1. Jalankan perintah berikut untuk membuat cadangan lokal.

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

2. Jalankan perintah berikut untuk menyimpan nama untuk cadangan.

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

3. Jalankan perintah berikut untuk mengunggah cadangan Anda ke bucket S3. Ganti *S3-Bucket* dengan nilai dari langkah 1 inLangkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks .

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

   Simpan `PUPPET_BACKUP` dan `S3_BUCKET` nilai. Anda akan mengimpor nilai-nilai tersebut ke EC2 instance baru.

Anda dapat keluar dari sesi SSH atau Session Manager.

Langkah 4: Luncurkan EC2 instance baru

Luncurkan EC2 instance baru dari EC2 konsol saat https://console.aws.amazon.com/ec2/menggunakan konfigurasi yang sama dengan server OpsWorks for Puppet Enterprise.

Nama parameter	Nilai
OS	Amazon Linux 2
Jenis instans	`InstanceType` Nilai dari langkah 2 dariLangkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks .
Nama key pair	`KeyPair`Nilai dari langkah 2 dariLangkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks .

Nama parameter	Nilai
VPC	VPC dari langkah 2 `SubnetIds` dari. Langkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks
Subnet	`SubnetIds` Dari langkah 2 dariLangkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks .
Pilih grup keamanan yang ada -> Grup keamanan umum	`SecurityGroupIds` Dari langkah 2 dariLangkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks .
Penyimpanan	Setidaknya 120 GB.
Profil instans IAM	`InstanceProfileArn` Dari langkah 2 dariLangkah 2: Dapatkan detail tentang server Puppet Enterprise Anda OpsWorks .

Jika Anda ingin membuat dan melampirkan IP Elastis ke instance baru, salin ID instance dari instance baru, dan selesaikan langkah-langkahnya(Opsional) Langkah 4.1: Buat dan pasang IP Elastis.

(Opsional) Langkah 4.1: Buat dan pasang IP Elastis

Dengan alamat IP Elastis, Anda dapat menutupi kegagalan suatu instans atau perangkat lunak dengan meremajakan secara cepat alamat ke instans lain di akun Anda.

Untuk membuat dan mengaitkan alamat IP Elastis

1. Masuk ke AWS Management Console dan buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/.
2. Pilih Elastis IPs.
3. Pilih Alokasi alamat IP elastis.
4. Dari halaman Allocate Elastic IP Address, pilih Alokasikan. Ini menciptakan IPv4 alamat Publik.
5. Salin IPv4 alamat yang dialokasikan.
6. Dari Tindakan, pilih Alamat IP Associate Elastic.
7. Misalnya, masukkan ID instance untuk instance baru.
8. Pilih Kaitkan.

Langkah 5: Instal Puppet Enterprise pada instance baru EC2

Gunakan SSH untuk terhubung ke EC2 instance baru. Anda dapat menggunakan Session Manager di EC2 konsol alih-alih 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
```

Anda dapat menjaga sesi SSH atau Session Manager tetap terbuka untuk langkah berikutnya.

Langkah 6: Kembalikan cadangan pada EC2 instance baru

```
# 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
```

Anda dapat mengakses konsol Puppet untuk EC2 instance yang dipulihkan di https://*Public IPv4 of the instance*. Anda dapat menemukan IPv4 DNS Publik di halaman detail instans di EC2 konsol. Kredensi login adalah kredensil yang sama yang Anda gunakan untuk mengakses server Puppet Enterprise Anda OpsWorks .

Anda dapat menjaga sesi SSH atau Session Manager tetap terbuka untuk langkah berikutnya.

Langkah 7: Konfigurasikan lisensi Boneka Anda

Ikuti langkah-langkah di situs web Puppet untuk mengonfigurasi lisensi Anda.

Anda dapat menjaga sesi SSH atau Session Manager tetap terbuka untuk langkah berikutnya.

Langkah 8: Migrasikan node Anda

Ada dua jenis domain yang didukung oleh server OpsWorks for Puppet Enterprise:

- BYODC (Bawa Domain dan Sertifikat Anda Sendiri)
- OpsWorks titik akhir

Langkah 8.1: Untuk BYODC (Bawa Domain dan Sertifikat Anda Sendiri)

Untuk node ini, yang perlu Anda lakukan adalah mengarahkan Domain Kustom di penyedia DNS Anda ke IPv4 DNS Publik atau IPv4 alamat Publik instance baru EC2 .

Langkah 8.2: Untuk titik akhir OpsWorks

Untuk OpsWorks endpoint, dokumentasi Puppet merekomendasikan untuk menghapus instalan agen Puppet pada node dan kemudian menginstal agen Puppet menggunakan server Puppet Enterprise yang baru dipulihkan.

> **Note**
>
> Meskipun Puppet tidak memiliki prosedur otomatis untuk memindahkan node agen, ada beberapa modul yang telah dipublikasikan oleh anggota komunitas Puppet di situs web Puppet Forge untuk menyelesaikan migrasi node otomatis. `pe_migrate`Modul-modul ini termasuk modul dan modul migrasi kedua oleh penulis yang berbeda. Modul di situs web

> Puppet Forge tidak didukung oleh Puppet atau OpsWorks kecuali secara eksplisit dicatat dalam modul Forge. Kami merekomendasikan untuk berhati-hati dengan modul-modul ini dan mengujinya sebelum digunakan secara luas.

Bagian berikut memberikan langkah-langkah untuk menghapus dan menginstal ulang agen Puppet pada instance Linux.

Topik

- Langkah 8.2.1: Salin uninstaller dari server Puppet
- Langkah 8.2.2: Unduh uninstaller dan jalankan pada node
- Langkah 8.2.3: Instal ulang agen Puppet pada node

Langkah 8.2.1: Salin uninstaller dari server Puppet

Sebelum Anda menghapus instalan agen, pastikan profil instance IAM node memberikan izin ReadOnly S3.

Jalankan perintah berikut untuk menyalin uninstaller dari server Puppet ke bucket S3.

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

Setelah menjalankan perintah, Anda dapat keluar dari sesi SSH atau Session Manager server Puppet.

Langkah 8.2.2: Unduh uninstaller dan jalankan pada node

Gunakan SSH untuk terhubung ke node. Anda dapat menggunakan Session Manager di EC2 konsol alih-alih SSH jika node adalah sebuah 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
```

Anda dapat menjaga sesi SSH atau Session Manager tetap terbuka untuk langkah berikutnya.

Langkah 8.2.3: Instal ulang agen Puppet pada node

Selesaikan langkah-langkah berikut untuk menginstal ulang agen Puppet pada node.

Topik

- Langkah 8.2.3.1: Instal agen Puppet dengan konfigurasi yang benar
- Langkah 8.2.3.2: Terima sertifikat di konsol Puppet
- Langkah 8.2.3.3: Periksa node ke server Puppet Enterprise

Langkah 8.2.3.1: Instal agen Puppet dengan konfigurasi yang benar

Jalankan perintah berikut untuk menginstal agen Puppet.

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

Anda dapat menjaga sesi SSH atau Session Manager tetap terbuka untuk langkah 8.2.2.3.

Langkah 8.2.3.2: Terima sertifikat di konsol Puppet

1. Pergi ke konsol server Puppet dihttps://*Public_IPv4_DNS*.
2. Pilih Sertifikat, lalu sertifikat yang tidak ditandatangani.
3. Pilih Terima untuk menandatangani sertifikat agen Wayang.

Langkah 8.2.3.3: Periksa node ke server Puppet Enterprise

Jalankan perintah berikut pada node untuk memeriksanya ke server.

```
puppet agent -t
```

Node sekarang harus terlihat di konsol server Puppet.

Langkah 9: Hapus server Puppet Enterprise Anda OpsWorks

Anda dapat menggunakan OpsWorks konsol atau AWS CLI menghapus server OpsWorks for Puppet Enterprise Anda.

Untuk menghapus server Anda menggunakan OpsWorks konsol

1. Masuk ke AWS Management Console dan buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Pilih server Puppet Enterprise dari panel navigasi.
3. Pada halaman server Puppet Enterprise, pilih server yang ingin Anda hapus.
4. Dari Actions, pilih Delete Puppet Enterprise server.

Untuk menghapus server Anda menggunakan AWS CLI

Jalankan perintah berikut.

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

Logging OpsWorks untuk Panggilan API Perusahaan Boneka dengan AWS CloudTrail

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

OpsWorks untuk Puppet Enterprise terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan dalam OpsWorks untuk Perusahaan Boneka. CloudTrail menangkap semua panggilan API OpsWorks untuk Puppet Enterprise sebagai acara, termasuk panggilan dari konsol OpsWorks for Puppet Enterprise dan dari panggilan kode ke OpsWorks for Puppet Enterprise. APIs Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara OpsWorks untuk Puppet Enterprise. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat OpsWorks untuk

Perusahaan Boneka, alamat IP dari mana permintaan itu dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

OpsWorks untuk Informasi Perusahaan Boneka di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di OpsWorks Perusahaan Wayang, kegiatan tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam sejarah Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan acara yang sedang berlangsung di AWS akun Anda, termasuk acara OpsWorks untuk Puppet Enterprise, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, ketika Anda membuat jejak di konsol, jejak ini diterapkan ke semua Wilayah. Trail mencatat peristiwa dari semua wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

Semua tindakan OpsWorks untuk Puppet Enterprise dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API OpsWorks for Puppet Enterprise. Misalnya, panggilan ke CreateServer, CreateBackup, dan DescribeServerstindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan tersebut dibuat dengan kredensial root atau pengguna IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

OpsWorks Memahami Entri Berkas Log Perusahaan Boneka

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log untuk `CreateServer` tindakan 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"
}
```

Pemecahan Masalah OpsWorks untuk Perusahaan Boneka

> **Important**
>
> AWS OpsWorks for Puppet Enterprise Layanan ini mencapai akhir masa pakai pada 31 Maret 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini berisi beberapa hal umum OpsWorks untuk masalah Perusahaan Boneka, dan solusi yang disarankan untuk masalah tersebut.

Topik

- Kiat pemecahan masalah umum
- Memecahkan masalah kesalahan tertentu
- Bantuan dan dukungan tambahan

Kiat pemecahan masalah umum

Jika Anda tidak dapat membuat atau bekerja dengan master Wayang, Anda dapat melihat pesan kesalahan atau log untuk membantu Anda memecahkan masalah. Tugas-tugas berikut menjelaskan tempat-tempat umum untuk memulai ketika Anda memecahkan masalah master boneka. Untuk informasi tentang kesalahan dan solusi tertentu, lihat Memecahkan masalah kesalahan tertentu bagian dari topik ini.

- Gunakan konsol OpsWorks for Puppet Enterprise untuk melihat pesan kesalahan jika master boneka gagal memulai. Pada halaman properti master boneka, pesan kesalahan yang terkait dengan peluncuran dan menjalankan server ditampilkan di bagian atas halaman. Kesalahan dapat berasal dari OpsWorks untuk Puppet Enterprise AWS CloudFormation,, atau Amazon EC2, layanan yang digunakan untuk membuat master boneka. Pada halaman properti, Anda juga dapat melihat peristiwa yang terjadi di server yang sedang berjalan, yang dapat berisi pesan peristiwa kegagalan.
- Untuk membantu menyelesaikan EC2 masalah, sambungkan ke instance server Anda dengan menggunakan SSH, dan lihat log. EC2 log contoh disimpan dalam `/var/log/aws/opsworks-`

`cm` direktori. Log ini menangkap output perintah sementara OpsWorks untuk Puppet Enterprise meluncurkan master Puppet.

Memecahkan masalah kesalahan tertentu

Topik

- Server dalam keadaan koneksi hilang
- Pembuatan server gagal dengan pesan “konfigurasi yang diminta saat ini tidak didukung”
- Tidak dapat membuat EC2 instance Amazon server
- Kesalahan peran layanan mencegah pembuatan server
- Batas alamat IP elastis terlampaui
- Asosiasi simpul tanpa pengawasan gagal
- Pemeliharaan sistem gagal

Server dalam keadaan koneksi hilang

Masalah: Status server ditampilkan sebagai Koneksi hilang.

Penyebab: Ini paling sering terjadi ketika entitas di luar AWS OpsWorks membuat perubahan pada server OpsWorks for Puppet Enterprise atau sumber daya pendukungnya. AWS OpsWorks tidak dapat terhubung ke server Puppet Enterprise di status yang hilang Koneksi untuk menangani tugas pemeliharaan seperti membuat cadangan, menerapkan tambalan sistem operasi, atau memperbarui Puppet. Akibatnya, server Anda mungkin kehilangan pembaruan penting, rentan terhadap masalah keamanan, atau tidak beroperasi seperti yang diharapkan.

Solusi: Coba langkah-langkah berikut untuk memulihkan koneksi server.

1. Pastikan bahwa peran layanan Anda memiliki semua izin yang diperlukan.

    a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk peran layanan yang digunakan server. Ini membuka peran layanan untuk dilihat di konsol IAM.

    b. Pada tab Izin, verifikasi yang `AWSOpsWorksCMServiceRole` ada di daftar Kebijakan izin. Jika tidak terdaftar, tambahkan kebijakan `AWSOpsWorksCMServiceRole` terkelola secara manual ke peran.

   c. Pada tab Hubungan kepercayaan, verifikasi bahwa peran layanan memiliki kebijakan kepercayaan yang mempercayai `opsworks-cm.amazonaws.com` layanan untuk mengambil peran atas nama Anda. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

2. Pastikan bahwa profil instans Anda memiliki semua izin yang diperlukan.

   a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk profil instance yang digunakan server. Ini membuka profil instance untuk dilihat di konsol IAM.

   b. Pada tab Izin, verifikasi itu `AmazonEC2RoleforSSM` dan keduanya `AWSOpsWorksCMInstanceProfileRole` ada di daftar Kebijakan izin. Jika salah satu atau keduanya tidak terdaftar, tambahkan kebijakan terkelola ini secara manual ke peran.

   c. Pada tab Hubungan kepercayaan, verifikasi bahwa peran layanan memiliki kebijakan kepercayaan yang mempercayai `ec2.amazonaws.com` layanan untuk mengambil peran atas nama Anda. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

3. Di EC2 konsol Amazon, pastikan Anda berada di wilayah yang sama dengan wilayah server OpsWorks for Puppet Enterprise, lalu restart EC2 instance yang digunakan server Anda.

   a. Pilih EC2 contoh yang diberi nama `aws-opsworks-cm-instance-`*`server-name`*.

   b. Pada menu status Instance, pilih Reboot instance.

   c. Biarkan hingga 15 menit agar server Anda restart dan sepenuhnya online.

4. Di konsol OpsWorks for Puppet Enterprise, pada halaman detail server, verifikasi bahwa status server sekarang sehat.

Jika status server masih Koneksi hilang setelah melakukan langkah-langkah sebelumnya, coba salah satu dari berikut ini.

- Ganti server dengan membuat yang baru dan menghapus yang asli. Jika data di server saat ini penting bagi Anda, pulihkan server dari cadangan terbaru, dan verifikasi data yang mutakhir sebelum menghapus server asli yang tidak responsif.
- Hubungi AWS dukungan.

Pembuatan server gagal dengan pesan “konfigurasi yang diminta saat ini tidak didukung”

Masalah: Anda mencoba membuat server Puppet Enterprise, tetapi pembuatan server gagal dengan pesan kesalahan yang mirip dengan “Konfigurasi yang diminta saat ini tidak didukung. Silakan periksa dokumentasi untuk konfigurasi yang didukung.”

Penyebab: Jenis instance yang tidak didukung mungkin telah ditentukan untuk master Puppet. Jika Anda memilih untuk membuat server Puppet di VPC yang memiliki penyewaan non-default, seperti server untuk instance khusus, semua instance di dalam VPC yang ditentukan juga harus memiliki penyewaan khusus atau host. Karena beberapa jenis instance, seperti t2, hanya tersedia dengan penyewaan default, tipe instance master Puppet mungkin tidak didukung oleh VPC yang ditentukan, dan pembuatan server gagal.

Solusi: Jika Anda memilih VPC yang memiliki tenancy non-default, gunakan tipe instans m4, yang dapat mendukung penyewaan khusus.

Tidak dapat membuat EC2 instance Amazon server

Masalah: Pembuatan server gagal dengan pesan kesalahan yang mirip dengan berikut ini: “Sumber daya berikut gagal dibuat: [EC2Instance]. Gagal menerima 1 sinyal sumber daya dalam durasi yang ditentukan.”

Penyebab: Ini kemungkinan besar karena EC2 instance tidak memiliki akses jaringan.

Solusi: Pastikan instans memiliki akses Internet keluar, dan agen AWS layanan dapat mengeluarkan perintah. Pastikan VPC Anda (VPC dengan subnet publik tunggal) mengaktifkan resolusi DNS, dan subnet Anda mengaktifkan pengaturan Auto-assign Public IP.

Kesalahan peran layanan mencegah pembuatan server

Masalah: Pembuatan server gagal dengan pesan kesalahan yang menyatakan, “Tidak diizinkan untuk melakukan sts:AssumeRole.”

Penyebab: Ini dapat terjadi ketika peran layanan yang Anda gunakan tidak memiliki izin yang memadai untuk membuat server baru.

Solusi: Buka konsol OpsWorks for Puppet Enterprise; gunakan konsol untuk menghasilkan peran layanan baru dan peran profil instance. Jika Anda lebih suka menggunakan peran layanan Anda sendiri, lampirkan kebijakan CMServicePeran AWSOps Kerja ke peran tersebut. Verifikasi bahwa

opsworks-cm.amazonaws.com terdaftar di antara layanan dalam hubungan Trust peran. Verifikasi bahwa peran layanan yang terkait dengan master Wayang memiliki kebijakan yang dikelola CMServicePeran AWSOps Karya terlampir.

Batas alamat IP elastis terlampaui

Masalah: Pembuatan server gagal dengan pesan kesalahan yang menyatakan, “Sumber daya berikut gagal dibuat: [EIP, EC2 Instance]. Pembuatan sumber daya dibatalkan, jumlah maksimum alamat telah tercapai.”

Penyebab: Ini terjadi ketika akun Anda telah menggunakan jumlah maksimum alamat Elastic IP (EIP). Batas alamat EIP default adalah lima.

Solusi: Anda dapat merilis alamat EIP yang ada atau menghapus alamat yang tidak digunakan secara aktif oleh akun Anda, atau Anda dapat menghubungi AWS Customer Support untuk meningkatkan batas alamat EIP yang terkait dengan akun Anda.

Asosiasi simpul tanpa pengawasan gagal

Masalah: Asosiasi EC2 node Amazon baru yang tidak dijaga, atau otomatis, gagal. Node yang seharusnya ditambahkan ke Puppet master tidak muncul di dashboard Puppet Enterprise.

Penyebab: Hal ini dapat terjadi ketika Anda tidak memiliki peran IAM yang disiapkan sebagai profil instance yang memungkinkan panggilan `opsworks-cm` API untuk berkomunikasi dengan instance baru EC2 .

Solusi: Lampirkan kebijakan ke profil EC2 instans Anda yang memungkinkan panggilan `DescribeNodeAssociationStatus` API `AssociateNode` dan berfungsi EC2, seperti yang dijelaskan dalamMenambahkan Node Secara Otomatis OpsWorks untuk Perusahaan Boneka.

Pemeliharaan sistem gagal

AWS OpsWorks CM melakukan pemeliharaan sistem mingguan untuk memastikan bahwa versi terbaru AWS dari Puppet Server, termasuk pembaruan keamanan, selalu berjalan di server OpsWorks for Puppet Enterprise. Jika, karena alasan apa pun, pemeliharaan sistem gagal, AWS OpsWorks CM memberi tahu Anda tentang kegagalan tersebut. Untuk informasi selengkapnya tentang pemeliharaan sistem, lihatPemeliharaan Sistem OpsWorks untuk Perusahaan Boneka.

Bagian ini menjelaskan kemungkinan alasan kegagalan dan menyarankan solusi.

Topik

- Peran layanan atau kesalahan profil instance mencegah pemeliharaan sistem

Peran layanan atau kesalahan profil instance mencegah pemeliharaan sistem

Masalah: Pemeliharaan sistem gagal dengan pesan kesalahan yang menyatakan, “Tidak diizinkan untuk melakukan sts: AssumeRole “, atau pesan kesalahan serupa tentang izin.

Penyebab: Ini dapat terjadi ketika peran layanan atau profil instance yang Anda gunakan tidak memiliki izin yang memadai untuk melakukan pemeliharaan sistem di server.

Solusi: Pastikan bahwa peran layanan dan profil instans Anda memiliki semua izin yang diperlukan.

1. Pastikan bahwa peran layanan Anda memiliki semua izin yang diperlukan.

    a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk peran layanan yang digunakan server. Ini membuka peran layanan untuk dilihat di konsol IAM.

    b. Pada tab Izin, verifikasi yang `AWSOpsWorksCMServiceRole` dilampirkan ke peran layanan. Jika `AWSOpsWorksCMServiceRole` tidak terdaftar, tambahkan kebijakan ini ke peran.

    c. Verifikasi bahwa opsworks-cm.amazonaws.com terdaftar di antara layanan dalam hubungan Trust peran. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

2. Pastikan bahwa profil instans Anda memiliki semua izin yang diperlukan.

    a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk profil instance yang digunakan server. Ini membuka profil instance untuk dilihat di konsol IAM.

    b. Pada tab Izin, verifikasi itu `AmazonEC2RoleforSSM` dan keduanya `AWSOpsWorksCMInstanceProfileRole` ada di daftar Kebijakan izin. Jika salah satu atau keduanya tidak terdaftar, tambahkan kebijakan terkelola ini secara manual ke peran.

    c. Pada tab Hubungan kepercayaan, verifikasi bahwa peran layanan memiliki kebijakan kepercayaan yang mempercayai `ec2.amazonaws.com` layanan untuk mengambil peran atas nama Anda. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

Bantuan dan dukungan tambahan

Jika Anda tidak melihat masalah spesifik Anda yang dijelaskan dalam topik ini, atau Anda telah mencoba saran dalam topik ini dan masih mengalami masalah, kunjungi AWS OpsWorks forum.

Anda juga dapat mengunjungi AWS Support Center. AWS Support Center adalah hub untuk membuat dan mengelola kasus AWS Support. AWS Support Center juga menyertakan tautan ke sumber daya bermanfaat lainnya, seperti forum, teknis FAQs, status kesehatan layanan, dan AWS Trusted Advisor.

AWS OpsWorks untuk Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks for Chef Automate memungkinkan Anda menjalankan server Chef Automate di AWS. Anda dapat menyediakan server Chef dalam beberapa menit, dan biarkan AWS OpsWorks for Chef Automate menangani operasi, pencadangan, restorasi, dan peningkatan perangkat lunaknya. AWS OpsWorks for Chef Automate membebaskan Anda untuk fokus pada tugas manajemen konfigurasi inti, alih-alih mengelola server Chef.

Server Chef Automate mengelola konfigurasi node di lingkungan Anda dengan menginstruksikan resep Chef `chef-client`mana yang akan dijalankan di node, menyimpan informasi tentang node, dan berfungsi sebagai repositori pusat untuk buku masak Chef Anda. AWS OpsWorks for Chef Automate menyediakan server Chef yang mencakup fitur premium Chef Automate: Chef Infra dan Chef. InSpec

AWS OpsWorks for Chef Automate Server berjalan pada instans Amazon Elastic Compute Cloud. AWS OpsWorks for Chef Automate server dikonfigurasi untuk menjalankan versi terbaru Amazon Linux (Amazon Linux 2). Untuk informasi tentang apa yang telah berubah dalam versi Chef Automate ini, lihat Catatan Rilis Otomatis Chef. Tabel berikut menjelaskan komponen Chef yang diinstal pada AWS OpsWorks for Chef Automate server.

Nama komponen	Deskripsi	Versi diinstal pada AWS OpsWorks for Chef Automate server
Chef Mengotomatisasi	Chef Automate adalah paket perangkat lunak server perusahaan yang menyediak an alur kerja otomatis untuk	2.0

Nama komponen	Deskripsi	Versi diinstal pada AWS OpsWorks for Chef Automate server
	penerapan berkelanjutan, dan wawasan tentang node terkelola di konsol manajemen berbasis web. Chef Automate memberikan otomatisasi infrastruktur dengan memasukkan Chef Infra, informasi keamanan dan kepatuhan serta penegakan hukum dengan memasukkan Chef InSpec, dan penyebaran otomatis dengan memasukkan Chef Habitat. Untuk informasi lebih lanjut tentang Chef Automate, lihat Chef Automate di situs web Chef.	
Koki Infra	Sebelumnya disebut Chef Server, Chef Infra Server menggunakan agen Chef Infra Client (`chef-client`) untuk terus menerapkan konfigurasi ke node yang dikelola untuk mempertahankan status yang diinginkan. Untuk informasi lebih lanjut tentang Infra, lihat Chef Infra di situs web Chef.	12.x

Nama komponen	Deskripsi	Versi diinstal pada AWS OpsWorks for Chef Automate server
Koki InSpec	Chef InSpec menjelaskan aturan keamanan dan kepatuhan yang dapat dibagi antara insinyur perangkat lunak, operasi, dan insinyur keamanan. Kepatuhan, keamanan, dan persyaratan kebijakan lainnya membentuk kerangka kerja untuk pengujian otomatis yang dapat dijalankan `chef-client` agen terhadap node yang dikelola, memastikan penegakan standar yang konsisten. Untuk informasi lebih lanjut tentang InSpec, lihat Chef InSpec di situs web Chef.	3.9.0

Versi minimum yang didukung `chef-client` pada node yang terkait dengan AWS OpsWorks for Chef Automate server adalah 13. x. Kami merekomendasikan menjalankan setidaknya 14.10.9, atau versi stabil `chef-client` terbaru.

Ketika versi minor baru dari perangkat lunak Chef tersedia, pemeliharaan sistem dirancang untuk memperbarui versi minor Chef Automate dan Chef Server di server secara otomatis, segera setelah melewati pengujian AWS. AWS melakukan pengujian ekstensif untuk memverifikasi bahwa peningkatan Chef siap produksi dan tidak mengganggu lingkungan pelanggan yang ada, sehingga mungkin ada kelambatan antara rilis perangkat lunak Chef dan ketersediaannya untuk aplikasi yang ada untuk server Chef Automate. OpsWorks Pemeliharaan sistem juga meningkatkan server Anda ke versi terbaru Amazon Linux.

Anda dapat menghubungkan komputer atau EC2 instans lokal yang menjalankan sistem operasi yang didukung dan memiliki akses jaringan ke AWS OpsWorks for Chef Automate server. Untuk daftar sistem operasi yang didukung untuk node yang ingin Anda kelola, lihat situs web Chef. Perangkat lunak `chef-client`agen diinstal pada node yang ingin Anda kelola dengan server Chef.

Topik

- Region Support untuk AWS OpsWorks for Chef Automate
- AWS OpsWorks untuk Chef Automate End of Life FAQs
- Tingkatkan AWS OpsWorks for Chef Automate Server ke Chef Automate 2
- Memulai dengan AWS OpsWorks for Chef Automate
- Membuat AWS OpsWorks for Chef Automate Server dengan menggunakan AWS CloudFormation
- Memperbarui AWS OpsWorks for Chef Automate Server untuk Menggunakan Domain Kustom
- Regenerasi starter kit untuk server AWS OpsWorks for Chef Automate
- Bekerja dengan Tag pada AWS OpsWorks for Chef Automate Sumber Daya
- Membuat cadangan dan Memulihkan AWS OpsWorks for Chef Automate Server
- Pemeliharaan Sistem di AWS OpsWorks for Chef Automate
- Pemindaian Kepatuhan di AWS OpsWorks for Chef Automate
- Memutuskan Node dari Server AWS OpsWorks for Chef Automate
- Menghapus AWS OpsWorks for Chef Automate Server
- Atur Ulang Kredensi Dasbor Otomatis Chef
- Pencatatan Panggilan AWS OpsWorks for Chef Automate API dengan AWS CloudTrail
- Pemecahan masalah AWS OpsWorks for Chef Automate

Region Support untuk AWS OpsWorks for Chef Automate

Endpoint regional berikut mendukung AWS OpsWorks for Chef Automate server. AWS OpsWorks for Chef Automate membuat sumber daya yang terkait dengan server Chef Anda, seperti profil instans, pengguna, dan peran layanan, di titik akhir regional yang sama dengan server Chef Anda. Server Chef Anda harus dalam VPC. Anda dapat menggunakan VPC yang Anda buat atau sudah miliki, atau menggunakan VPC default.

- Wilayah AS Timur (Ohio)
- Wilayah AS Timur (Virginia Utara)

- Wilayah AS Barat (California Utara)
- Wilayah US West (Oregon)
- Wilayah Asia Pasifik (Tokyo)
- Wilayah Asia Pasifik (Singapore)
- Wilayah Asia Pasifik (Sydney)
- Wilayah Eropa (Frankfurt)
- Wilayah Eropa (Irlandia)

AWS OpsWorks untuk Chef Automate End of Life FAQs

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif.

Topik

- Bagaimana pengguna yang ada akan terpengaruh oleh End of Life ini?
- Apa yang terjadi pada server saya jika saya tidak mengambil tindakan apa pun?
- Alternatif apa yang bisa saya transisi?
- Apakah layanan ini masih menerima pelanggan baru?
- Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama?
- Tingkat dukungan teknis apa yang tersedia?
- Saya adalah pelanggan saat ini OpsWorks untuk Chef Automate dan saya perlu meluncurkan server di akun yang sebelumnya tidak menggunakan layanan. Apakah saya bisa melakukan ini?
- Apakah akan ada rilis fitur utama selama tahun depan?

Bagaimana pengguna yang ada akan terpengaruh oleh End of Life ini?

Pelanggan yang ada tidak akan terpengaruh hingga 5 Mei 2024, tanggal Akhir Kehidupan OpsWorks untuk Chef Automate. Setelah tanggal End of Life, pelanggan tidak akan lagi dapat mengelola server mereka menggunakan OpsWorks konsol atau API.

Apa yang terjadi pada server saya jika saya tidak mengambil tindakan apa pun?

Mulai 5 Mei 2024, Anda tidak akan lagi dapat mengelola server Anda menggunakan OpsWorks konsol atau API. Pada saat itu, kami akan berhenti melakukan fungsi manajemen yang sedang berlangsung untuk server Anda seperti pencadangan atau pemeliharaan. Untuk membatasi dampaknya kepada pelanggan, kami akan membiarkan semua EC2 instance berjalan yang mencadangkan server Chef Automate, tetapi lisensi mereka tidak lagi berlaku karena penggunaan tidak lagi tercakup (atau ditagih) berdasarkan perjanjian layanan OpsWorks for Chef Automate dengan Chef. Anda harus menghubungi Chef untuk mendapatkan lisensi baru. Saat Anda menghubungi Chef, pastikan untuk memberi tahu mereka bahwa Anda sudah ada OpsWorks untuk pelanggan Chef Automate dan Anda sedang beralih dari. OpsWorks

Alternatif apa yang bisa saya transisi?

AWS dan Progress Chef merekomendasikan Anda untuk bermigrasi ke penawaran Chef SaaS baru mereka sehingga Anda dapat terus mendapatkan keuntungan dari layanan Chef Automate yang dikelola sepenuhnya. Untuk memulai dengan Chef SaaS, Anda dapat menghubungi Chef untuk mendapatkan dokumentasi tentang cara menyiapkan akun Chef SaaS dan mentransisikan data dan node Anda.

Jika Chef SaaS tidak akan memenuhi kebutuhan Anda karena Anda lebih suka menjalankan Chef Automate pada EC2 instans di AWS akun yang Anda kontrol, Chef menyediakan beberapa opsi termasuk model AWS Marketplace Bring Your Own License (BYOL) dan hosting mandiri. EC2 Anda dapat menghubungi Progress Chef untuk informasi lebih lanjut tentang cara menjalankan transisi semacam itu.

Apakah layanan ini masih menerima pelanggan baru?

Tidak. AWS OpsWorks karena Chef Automate tidak lagi menerima pelanggan baru.

Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama?

Ya. API dan Konsol akan mencapai End of Life dan tidak dapat digunakan pada 5 Mei 2024 secara keseluruhan. Wilayah AWS Untuk informasi tentang Wilayah AWS tempat AWS OpsWorks Chef Automate tersedia, lihat Daftar Layanan AWS Regional.

Tingkat dukungan teknis apa yang tersedia?

AWS akan terus memberikan tingkat dukungan yang sama OpsWorks untuk Chef Automate yang dimiliki pelanggan saat ini hingga tanggal Akhir Kehidupan. Jika Anda memiliki pertanyaan atau masalah, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium. Untuk dukungan transisi, kami menyarankan agar pelanggan menghubungi Progress Chef.

Saya adalah pelanggan saat ini OpsWorks untuk Chef Automate dan saya perlu meluncurkan server di akun yang sebelumnya tidak menggunakan layanan. Apakah saya bisa melakukan ini?

Umumnya tidak, kecuali ada keadaan luar biasa untuk melakukannya. Jika Anda memiliki situasi khusus, hubungi AWS Dukungan Tim di AWS Re:Post atau melalui AWS Dukungan Premium dengan rincian dan pembenaran untuk ini dan kami akan meninjau permintaan Anda.

Apakah akan ada rilis fitur utama selama tahun depan?

Tidak. Karena layanan ini mencapai End of Life, kami tidak akan merilis fitur baru apa pun. Namun, kami akan terus melakukan peningkatan keamanan dan mengelola server seperti yang diharapkan hingga tanggal End of Life.

Tingkatkan AWS OpsWorks for Chef Automate Server ke Chef Automate 2

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Prasyarat untuk Upgrade ke Chef Automate 2

Sebelum memulai, pastikan Anda memahami fitur-fitur baru yang ditambahkan Chef Automate 2, dan fitur yang tidak didukung Chef Automate 2. Untuk informasi tentang fitur baru dan yang tidak didukung di Chef Automate 2, lihat dokumentasi Chef Automate 2 di situs web Chef.

Server yang menjalankan Chef Automate 1 harus memiliki setidaknya satu pemeliharaan yang berhasil dijalankan setelah 1 November 2019 agar memenuhi syarat untuk peningkatan.

Seperti halnya operasi pemeliharaan di AWS OpsWorks for Chef Automate server Anda, server offline selama peningkatan. Anda harus merencanakan hingga tiga jam downtime selama proses upgrade.

Anda memerlukan kredensi masuk untuk server ini untuk situs web dasbor Chef Automate. Ketika pemutakhiran selesai, Anda harus masuk ke dasbor Chef Automate dan memverifikasi bahwa node dan informasi konfigurasi Anda tidak berubah.

> **Important**
>
> Ketika Anda siap untuk meningkatkan AWS OpsWorks for Chef Automate server Anda ke Chef Automate 2, gunakan hanya instruksi di sini untuk meningkatkan. Karena AWS OpsWorks for Chef Automate mengotomatiskan banyak proses pemutakhiran, seperti pembuatan cadangan, jangan ikuti petunjuk peningkatan di situs web Chef.

Tentang Proses Upgrade

Selama proses upgrade, server Anda dicadangkan sebelum memulai upgrade dan setelah menyelesaikan upgrade. Cadangan berikut dibuat:

- Cadangan server saat masih menjalankan Chef Automate 1 (versi 12.17.33).
- Cadangan server setelah upgrade selesai dan server menjalankan Chef Automate 2 (versi 2019-08).

Proses pemutakhiran menghentikan EC2 instance Amazon yang digunakan server saat menjalankan Chef Automate 1. Sebuah instance baru dibuat untuk menjalankan server Chef Automate 2.

Tingkatkan ke Chef Automate 2 (Konsol)

1. Masuk ke AWS Management Console dan buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di panel navigasi kiri, pilih AWS OpsWorks for Chef Automate.
3. Pilih server untuk melihat halaman propertinya. Spanduk biru di bagian atas halaman harus menunjukkan apakah server memenuhi syarat untuk meningkatkan ke Chef Automate 2.

   > **Note**
   >
   > Server yang menjalankan Chef Automate 1 harus memiliki setidaknya satu pemeliharaan yang berhasil dijalankan setelah 1 November 2019 agar memenuhi syarat untuk peningkatan.

4. Jika server memenuhi syarat untuk upgrade, pilih Mulai upgrade.
5. Biarkan hingga tiga jam untuk upgrade. Selama proses upgrade, halaman properti menampilkan status server sebagai Under maintenance.
6. Ketika pemutakhiran selesai, halaman properti menampilkan dua pesan berikut: Berhasil ditingkatkan ke Otomatisasi 2 dan Pemeliharaan berhasil diselesaikan. Status server harus SEHAT.
7. Masuk ke dasbor Chef Automate dengan kredensi Anda yang ada, dan verifikasi bahwa node Anda melapor dengan benar.

Tingkatkan ke Chef Automate 2 (CLI)

1. (Opsional) Jika Anda tidak yakin AWS OpsWorks for Chef Automate server mana yang memenuhi syarat untuk upgrade, jalankan perintah berikut. Pastikan untuk menambahkan `--region` parameter jika Anda ingin membuat daftar AWS OpsWorks for Chef Automate server di Wilayah AWS yang berbeda dari Wilayah AWS default Anda.

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

   Dalam hasil, cari nilai `true` untuk atribut`CHEF_MAJOR_UPGRADE_AVAILABLE`. Ini menunjukkan bahwa server memenuhi syarat untuk meningkatkan ke Chef Automate 2. Catat nama-nama AWS OpsWorks for Chef Automate server yang memenuhi syarat untuk upgrade.

2. Jalankan perintah berikut, ganti *server_name* dengan nama AWS OpsWorks for Chef Automate server. Untuk meningkatkan ke Chef Automate 2 alih-alih melakukan pemeliharaan sistem rutin, tambahkan atribut `CHEF_MAJOR_UPGRADE` engine, seperti yang ditunjukkan pada perintah. Tambahkan `--region` parameter jika server target tidak ada di Wilayah AWS default Anda. Anda hanya dapat meng-upgrade satu server per perintah.

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

   Jika AWS OpsWorks for Chef Automate tidak dapat memutakhirkan server karena alasan apa pun, perintah ini menghasilkan pengecualian validasi.

3. Biarkan hingga tiga jam untuk peningkatan. Anda dapat memeriksa status pemutakhiran secara berkala dengan menjalankan perintah berikut.

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

   Dalam hasilnya, cari `Status` nilainya. A `Status` `UNDER_MAINTENANCE` berarti bahwa upgrade masih dalam proses. Upgrade yang berhasil mengembalikan pesan yang mirip dengan berikut ini.

```
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
```

   Jika pemutakhiran tidak berhasil, AWS OpsWorks for Chef Automate secara otomatis memutar kembali server Anda ke Chef Automate 1.

   Jika upgrade berhasil tetapi server tidak berfungsi sama seperti sebelum upgrade (misalnya, jika node terkelola tidak melaporkan), Anda dapat memutar server kembali secara manual. Untuk informasi rollback manual, lihat. Kembalikan AWS OpsWorks for Chef Automate Server ke Chef Automate 1 (CLI)

Kembalikan AWS OpsWorks for Chef Automate Server ke Chef Automate 1 (CLI)

Jika proses upgrade gagal, AWS OpsWorks for Chef Automate otomatis memutar server Anda kembali ke Chef Automate 1. Jika upgrade berhasil tetapi server tidak berfungsi sama seperti

sebelum upgrade, Anda dapat memutar AWS OpsWorks for Chef Automate server Anda kembali ke Chef Automate 1 secara manual dengan menggunakan. AWS CLI

1. Jalankan perintah berikut untuk menunjukkan cadangan terakhir yang dilakukan di server Anda sebelum Anda mencoba upgrade. `BackupId` Tambahkan `--region` parameter jika server Anda berada di Wilayah AWS yang berbeda dari Wilayah AWS default Anda.

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

   Backup IDs dalam format*ServerName-yyyyMMddHHmmssSSS*. Cari properti Chef Automate 1 berikut di hasil.

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

2. Jalankan perintah berikut, menggunakan ID cadangan yang Anda kembalikan pada langkah 1 sebagai nilai`--backup-id`.

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

   Biarkan antara 20 menit dan tiga jam untuk memulihkan server, tergantung pada jumlah data yang Anda simpan di server. Selama operasi pemulihan, server Anda memiliki status`RESTORING`. Status ini ditampilkan pada halaman properti server di AWS Management Console, dan dikembalikan dalam hasil describe-servers perintah.

3. Setelah restorasi selesai, konsol menampilkan pesan, Pulihkan selesai dengan sukses. AWS OpsWorks for Chef Automate Server Anda sedang online, dan sama seperti sebelum Anda memulai proses upgrade.

Lihat Juga

- Pemeliharaan Sistem di AWS OpsWorks for Chef Automate
- Memulihkan AWS OpsWorks for Chef Automate Server dari Cadangan
- DescribeServers di Referensi API AWS OpsWorks
- StartMaintenance di Referensi API AWS OpsWorks

Memulai dengan AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks for Chef Automate memungkinkan Anda menjalankan server Chef Automate di AWS. Anda dapat menyediakan server Chef dalam waktu sekitar 15 menit.

Mulai 3 Mei 2021, AWS OpsWorks for Chef Automate menyimpan beberapa atribut server Chef Automate di AWS Secrets Manager. Untuk informasi selengkapnya, lihat Integrasi dengan AWS Secrets Manager.

Panduan berikut membantu Anda membuat server Chef pertama Anda di. AWS OpsWorks for Chef Automate

Prasyarat

Sebelum Anda mulai, Anda harus menyelesaikan prasyarat berikut.

Topik

- Siapkan VPC
- Prasyarat untuk Menggunakan Domain Kustom (Opsional)
- Mengatur Pasangan EC2 Kunci (Opsional)

Siapkan VPC

AWS OpsWorks for Chef Automate Server Anda harus beroperasi di Amazon Virtual Private Cloud. Anda dapat menambahkannya ke VPC yang ada, menggunakan VPC default, atau membuat VPC baru untuk memuat server. Untuk informasi tentang Amazon VPC dan cara membuat VPC baru, lihat Panduan Memulai VPC Amazon.

Jika Anda membuat VPC Anda sendiri, atau menggunakan yang sudah ada, VPC harus memiliki pengaturan atau properti berikut.

- VPC harus memiliki setidaknya satu subnet.

  Jika AWS OpsWorks for Chef Automate server Anda dapat diakses publik, buat subnet publik, dan aktifkan Auto-assign IP publik.
- Resolusi DNS harus diaktifkan.
- Pada subnet, aktifkan Auto-assign IP publik.

Prasyarat untuk Menggunakan Domain Kustom (Opsional)

Anda dapat mengatur server Chef Automate di domain Anda sendiri, menentukan titik akhir publik dalam domain khusus untuk digunakan sebagai titik akhir server Anda. Saat Anda menggunakan domain khusus, semua hal berikut diperlukan, seperti yang dijelaskan secara rinci di bagian ini.

Topik

- Mengatur Domain Kustom
- Dapatkan Sertifikat
- Dapatkan Kunci Pribadi

Mengatur Domain Kustom

Untuk menjalankan server Chef Automate Anda di domain kustom Anda sendiri, Anda memerlukan titik akhir publik server, seperti. `https://aws.my-company.com` Jika Anda menentukan domain kustom, Anda juga harus memberikan sertifikat dan kunci pribadi, seperti yang dijelaskan di bagian sebelumnya.

Untuk mengakses server setelah Anda membuatnya, tambahkan catatan DNS CNAME di layanan DNS pilihan Anda. Catatan ini harus mengarahkan domain kustom ke titik akhir (nilai `Endpoint` atribut server) yang dihasilkan oleh proses pembuatan server Chef Automate. Anda tidak dapat mengakses server menggunakan nilai `Endpoint` jika server menggunakan domain kustom.

Dapatkan Sertifikat

Untuk mengatur server Chef Automate Anda di domain kustom Anda sendiri, Anda memerlukan sertifikat HTTPS berformat PEM. Ini bisa berupa sertifikat tunggal yang ditandatangani sendiri,

atau rantai sertifikat. Saat Anda menyelesaikan alur kerja server Create Chef Automate, jika Anda menentukan sertifikat ini, Anda juga harus menyediakan domain khusus dan kunci pribadi.

Berikut ini adalah persyaratan untuk nilai sertifikat:

- Anda dapat memberikan sertifikat yang ditandatangani sendiri, kustom, atau rantai sertifikat penuh.
- Sertifikat harus merupakan sertifikat X509 yang valid, atau rantai sertifikat dalam format PEM.
- Sertifikat harus valid pada saat diunggah. Sertifikat tidak dapat digunakan sebelum periode validitasnya dimulai (tanggal `NotBefore` sertifikat) atau setelah kedaluwarsa (tanggal `NotAfter` sertifikat).
- Nama umum sertifikat atau nama alternatif subjek (SANs), jika ada, harus sesuai dengan nilai domain kustom.
- Sertifikat harus sesuai dengan nilai bidang kunci pribadi kustom.

Dapatkan Kunci Pribadi

Untuk mengatur server Chef Automate Anda di domain kustom Anda sendiri, Anda memerlukan kunci pribadi dalam format PEM untuk menghubungkan ke server dengan menggunakan HTTPS. Kunci privat tidak boleh dienkripsi; tidak dapat dilindungi oleh kata sandi atau frasa sandi. Jika Anda menentukan kunci pribadi khusus, Anda juga harus menyediakan domain khusus dan sertifikat.

Mengatur Pasangan EC2 Kunci (Opsional)

Koneksi SSH tidak diperlukan atau direkomendasikan untuk manajemen khas server Chef; Anda dapat menggunakan `knife`perintah untuk melakukan sebagian besar tugas manajemen di server Chef Anda.

EC2 Key pair diperlukan untuk terhubung ke server Anda dengan menggunakan SSH jika Anda kehilangan atau ingin mengubah kata sandi masuk untuk dasbor Chef Automate. Anda dapat menggunakan key pair yang sudah ada, atau membuat key pair baru. Untuk informasi selengkapnya tentang cara membuat EC2 key pair baru, lihat Amazon EC2 Key Pairs.

Jika Anda tidak memerlukan EC2 key pair, Anda siap membuat server Chef.

Buat Server Otomatis Koki

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

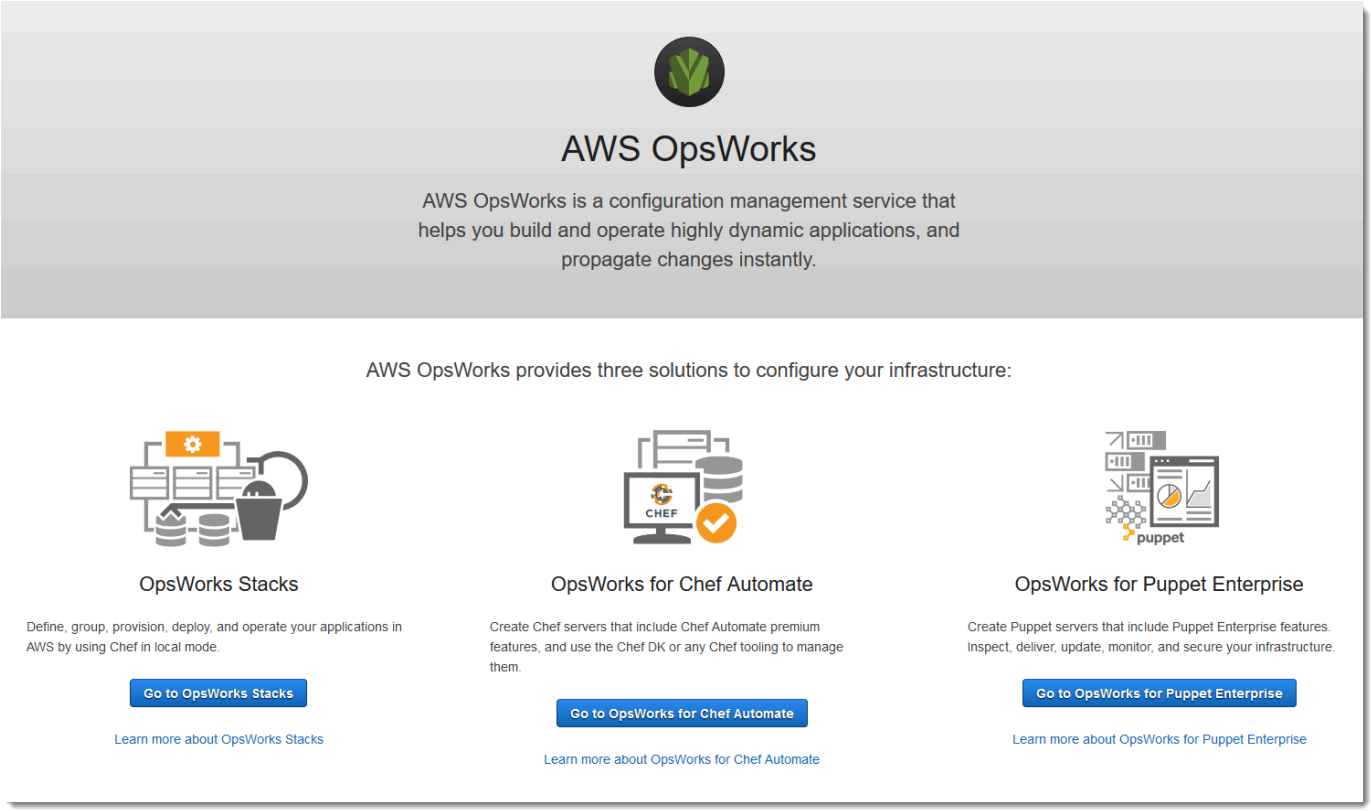
Anda dapat membuat server Chef dengan menggunakan AWS OpsWorks for Chef Automate konsol, atau file AWS CLI.

Topik

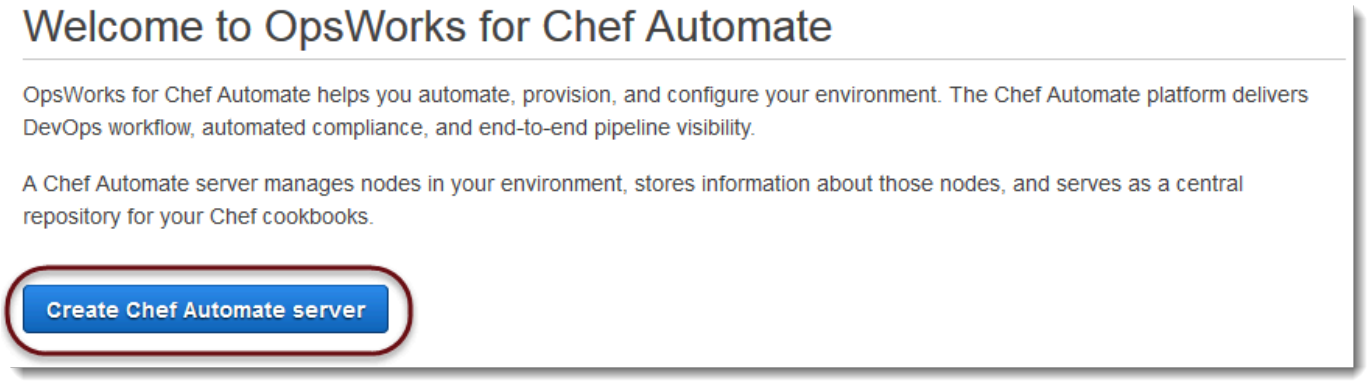
- Buat server Chef Automate di AWS Management Console
- Buat server Chef Automate dengan menggunakan AWS CLI

Buat server Chef Automate di AWS Management Console

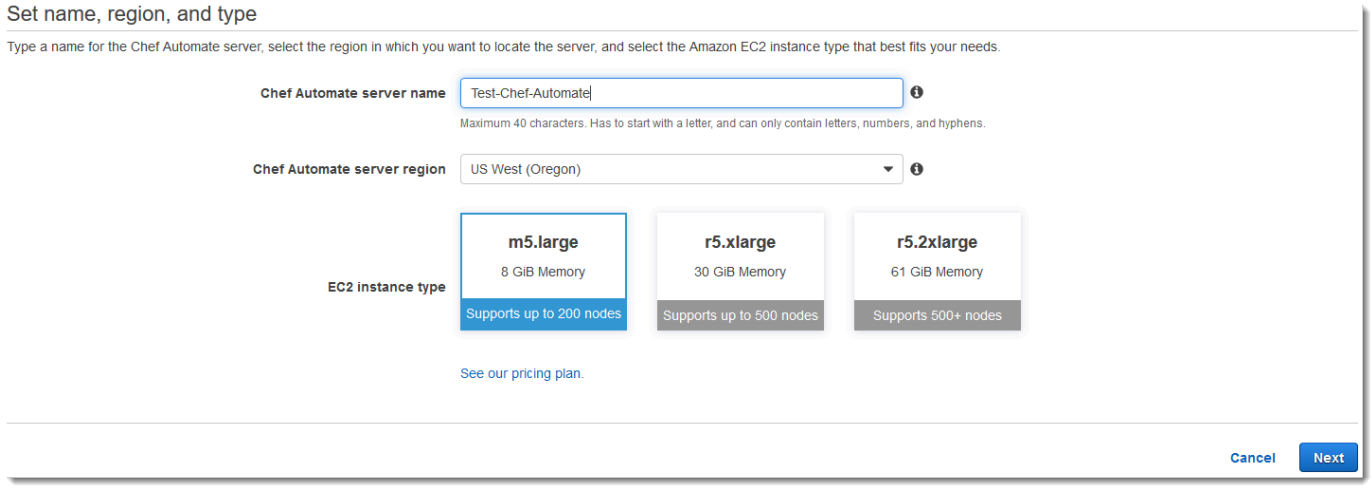
1. Masuk ke AWS Management Console dan buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di AWS OpsWorks halaman beranda, pilih Buka OpsWorks untuk Chef Automate.



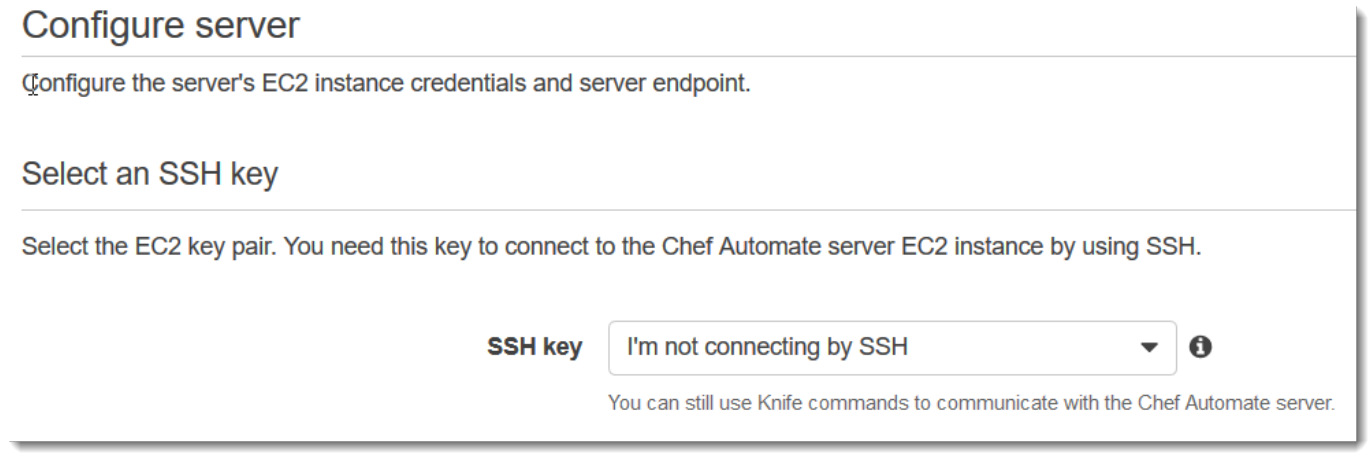
3. Pada AWS OpsWorks for Chef Automate halaman beranda, pilih Create Chef Automate server.



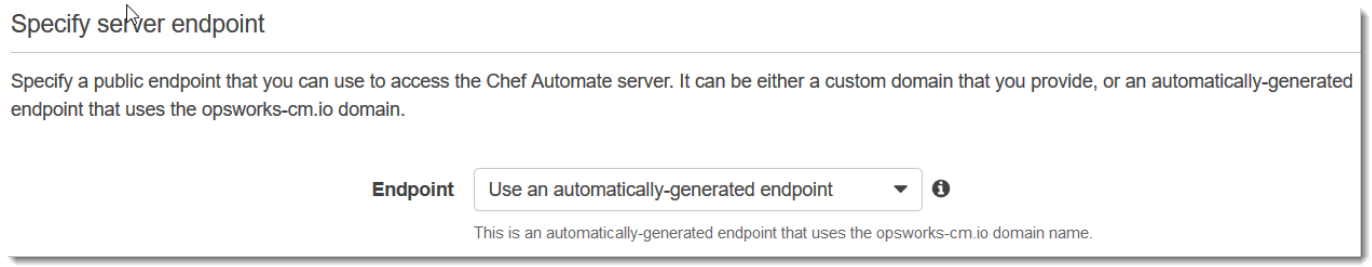
4. Pada halaman Set name, region, dan type, tentukan nama untuk server Anda. Nama server chef dapat maksimal 40 karakter, dan hanya dapat berisi karakter alfanumerik dan tanda hubung. Pilih wilayah yang didukung, lalu pilih jenis instance yang mendukung jumlah node yang ingin Anda kelola. Anda dapat mengubah jenis instance setelah server Anda dibuat, jika diperlukan. Untuk panduan ini, kami membuat tipe instance m5.large di Wilayah AS Barat (Oregon). Pilih Berikutnya.



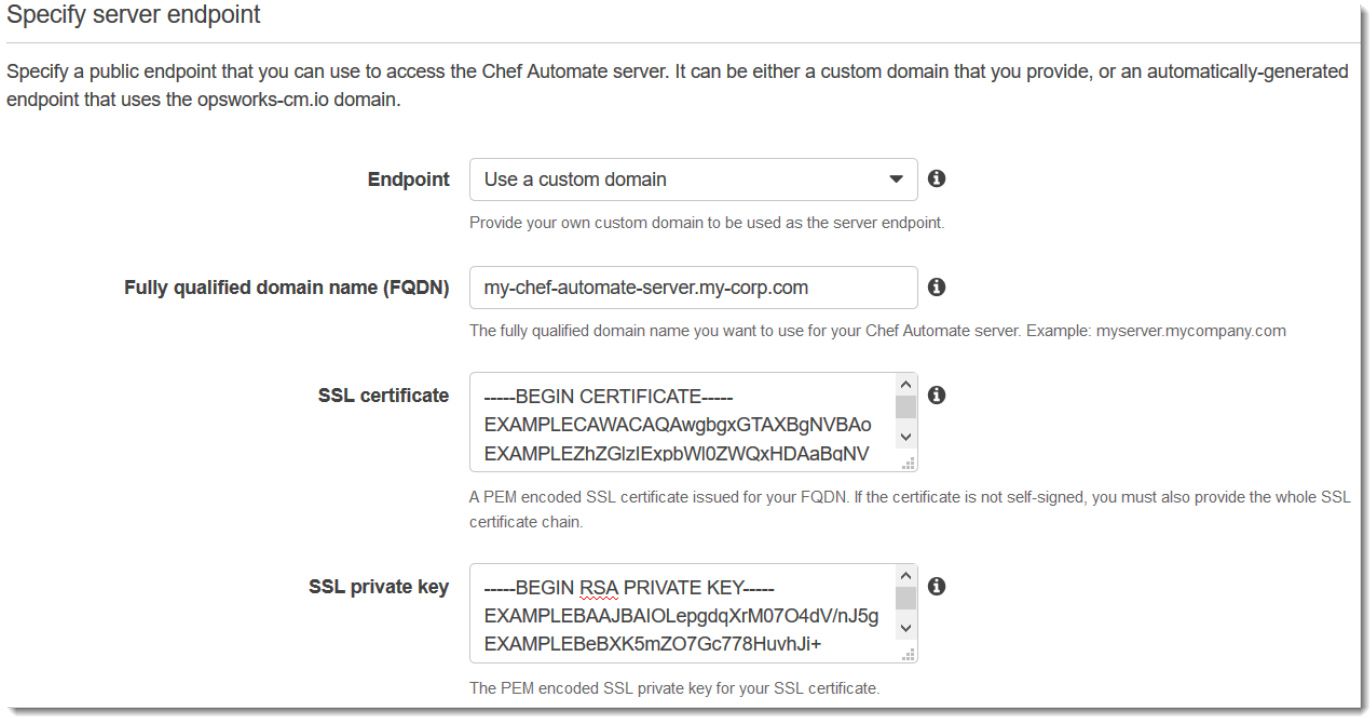
5. Pada halaman Configure server, tinggalkan pilihan default di daftar drop-down kunci SSH, kecuali jika Anda ingin menentukan nama key pair.



6. Untuk Tentukan titik akhir server, biarkan default, Gunakan titik akhir yang dihasilkan secara otomatis lalu pilih Berikutnya, kecuali jika Anda ingin server Anda berada di domain kustom Anda sendiri. Untuk mengonfigurasi domain khusus, lanjutkan ke langkah berikutnya.



7. Untuk menggunakan domain kustom, untuk Menentukan titik akhir server, pilih Gunakan domain kustom dari daftar drop-down.



a. Untuk nama domain yang sepenuhnya memenuhi syarat (FQDN), tentukan FQDN. Anda harus memiliki nama domain yang ingin Anda gunakan.

b. Untuk sertifikat SSL, tempel di seluruh sertifikat berformat PEM, dimulai dengan dan diakhiri dengan`-----BEGIN CERTIFICATE-----`. `-----END CERTIFICATE-----` Subjek sertifikat SSL harus sesuai dengan FQDN yang Anda masukkan pada langkah sebelumnya.

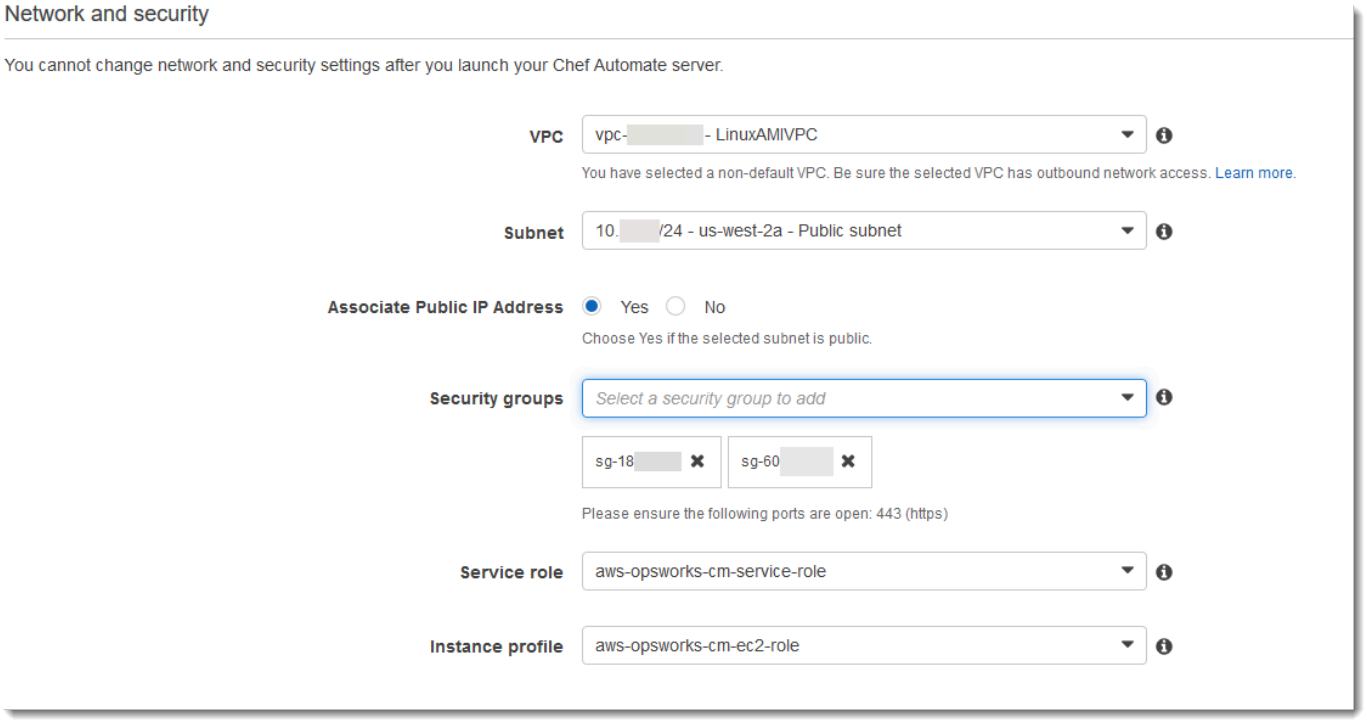
c. Untuk kunci pribadi SSL, tempel di seluruh kunci pribadi RSA, dimulai dengan `-----BEGIN RSA PRIVATE KEY-----` dan diakhiri dengan. `-----END RSA PRIVATE KEY-----` Kunci pribadi SSL harus sesuai dengan kunci publik dalam sertifikat SSL yang Anda masukkan pada langkah sebelumnya. Pilih Berikutnya.

8. Pada halaman Konfigurasi Pengaturan Lanjut, di area Jaringan dan Keamanan, pilih VPC, subnet, dan satu atau beberapa grup keamanan. Berikut ini adalah persyaratan untuk VPC Anda:

   - VPC harus memiliki setidaknya satu subnet publik.
   - Resolusi DNS harus diaktifkan.
   - Tetapkan IP publik secara otomatis harus diaktifkan pada subnet publik.

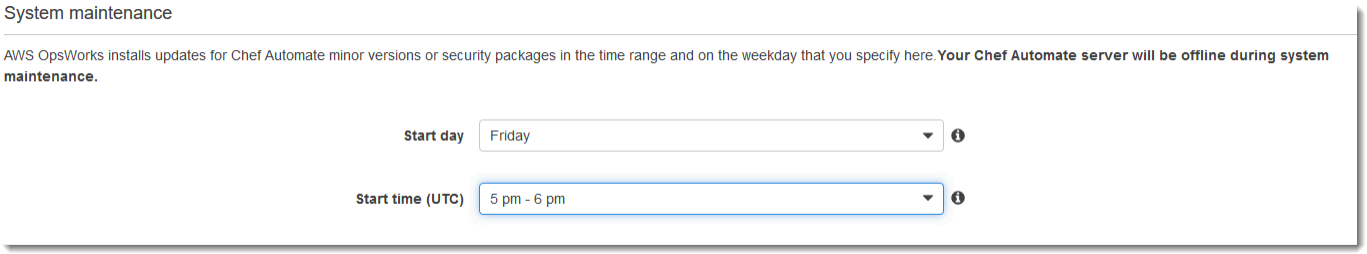
   AWS OpsWorks dapat menghasilkan grup keamanan, peran layanan, dan profil instance untuk Anda, jika Anda belum memiliki yang ingin Anda gunakan. Server Anda dapat menjadi anggota

dari beberapa grup keamanan. Anda tidak dapat mengubah pengaturan jaringan dan keamanan untuk server Chef setelah Anda meninggalkan halaman ini.

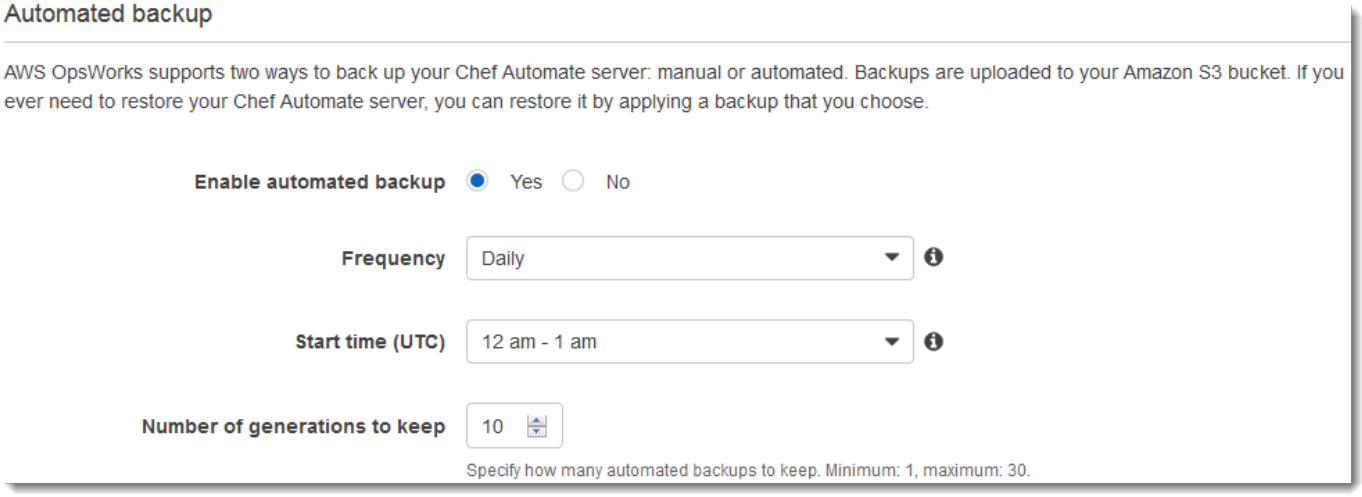


9. Di bagian Pemeliharaan sistem, atur hari dan jam yang Anda inginkan untuk memulai pemeliharaan sistem. Karena Anda harus mengharapkan server offline selama pemeliharaan sistem, pilih waktu permintaan server rendah dalam jam kantor reguler. Node yang terhubung memasuki `pending-server` status sampai pemeliharaan selesai.

   Jendela pemeliharaan diperlukan. Anda dapat mengubah hari mulai dan waktu kemudian dengan menggunakan AWS Management Console, AWS CLI, atau APIs.



10. Konfigurasikan cadangan. Secara default, backup otomatis diaktifkan. Tetapkan frekuensi dan jam yang diinginkan agar pencadangan otomatis dimulai, dan atur jumlah generasi cadangan yang akan disimpan di Amazon Simple Storage Service. Maksimal 30 cadangan disimpan; ketika maksimum tercapai, AWS OpsWorks for Chef Automate hapus cadangan tertua untuk memberi ruang bagi yang baru.



11. (Opsional) Di Tag, tambahkan tag ke server dan sumber daya terkait, seperti EC2 contoh, alamat IP Elastis, grup keamanan, bucket S3, dan cadangan. Untuk informasi selengkapnya tentang menandai AWS OpsWorks for Chef Automate server, lihatBekerja dengan Tag pada AWS OpsWorks for Chef Automate Sumber Daya.
12. Setelah selesai mengonfigurasi pengaturan lanjutan, pilih Berikutnya.
13. Pada halaman Tinjau, tinjau pilihan Anda. Ketika Anda siap untuk membuat server, pilih Launch.

    Sementara Anda menunggu AWS OpsWorks untuk membuat server Chef Anda, lanjutkan ke Konfigurasikan Server Chef Menggunakan Starter Kit dan unduh Starter Kit dan kredenal dasbor Chef Automate. Jangan menunggu sampai server Anda online untuk mengunduh item ini.

    Ketika pembuatan server selesai, server Chef Anda tersedia di AWS OpsWorks for Chef Automate halaman beranda, dengan status online. Setelah server online, dasbor Chef Automate tersedia di domain server, di URL dalam format berikut:`https://`*`your_server_name`*`-random.region.opsworks-cm.io`.

Buat server Chef Automate dengan menggunakan AWS CLI

Membuat AWS OpsWorks for Chef Automate server dengan menjalankan AWS CLI perintah berbeda dari membuat server di konsol. Di konsol, AWS OpsWorks buat peran layanan dan grup keamanan untuk Anda, jika Anda tidak menentukan yang sudah ada yang ingin Anda gunakan. Dalam AWS CLI, AWS OpsWorks dapat membuat grup keamanan untuk Anda jika Anda tidak menentukan satu, tetapi tidak secara otomatis membuat peran layanan; Anda harus memberikan peran layanan ARN sebagai bagian dari perintah Anda`create-server`. Di konsol, saat AWS OpsWorks membuat server Chef Automate Anda, Anda mengunduh starter kit Chef Automate dan kredenal masuk untuk dasbor Chef

Automate. Karena Anda tidak dapat melakukan ini ketika Anda membuat AWS OpsWorks for Chef Automate server dengan menggunakan AWS CLI, Anda menggunakan utilitas pemrosesan JSON untuk mendapatkan kredensi masuk dan starter kit dari hasil `create-server` perintah setelah server baru AWS OpsWorks for Chef Automate Anda online. Atau, Anda dapat menghasilkan satu set kredenal masuk baru dan starter kit baru di konsol setelah AWS OpsWorks for Chef Automate server baru Anda online.

Jika komputer lokal Anda belum menjalankan AWS CLI, unduh dan instal AWS CLI dengan mengikuti petunjuk penginstalan di Panduan Pengguna Antarmuka Baris Perintah AWS. Bagian ini tidak menjelaskan semua parameter yang dapat Anda gunakan dengan `create-server` perintah. Untuk informasi selengkapnya tentang `create-server` parameter, lihat `create-server`di AWS CLI Referensi.

1. Pastikan untuk melengkapi prasyarat, terutamaSiapkan VPC, atau pastikan Anda memiliki VPC yang ingin Anda gunakan. Untuk membuat server Chef Automate Anda, Anda memerlukan subnet ID.
2. Secara opsional, buat kunci penting Chef dengan menggunakan OpenSSL, dan simpan kunci ke file yang aman dan nyaman di komputer lokal Anda. Kunci penting secara otomatis dihasilkan sebagai bagian dari proses pembuatan server jika Anda tidak memberikannya dalam perintah. `create-server` Jika Anda ingin melewati langkah ini, Anda bisa mendapatkan kunci penting Chef Automate dari hasil perintah. `create-server` Jika Anda memilih untuk menghasilkan kunci penting menggunakan perintah berikut, pastikan untuk menyertakan `-pubout` parameter, karena nilai kunci penting Chef Automate adalah separuh publik dari key pair RSA. Untuk informasi lebih lanjut, lihat Langkah 6.

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

3. Buat peran layanan dan profil instance.
4. Temukan dan salin ARNs peran layanan di akun Anda.

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

   Dalam hasil `list-roles` perintah, cari entri ARN peran layanan yang menyerupai berikut ini. Catat peran layanan ARNs. Anda memerlukan nilai-nilai ini untuk membuat server Chef Automate Anda.

```
{
    "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. Temukan dan salin profil instans di akun Anda. ARNs

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

Dalam hasil `list-instance-profiles` perintah, cari contoh entri ARN profil yang menyerupai berikut ini. Buat catatan profil instance ARNs. Anda memerlukan nilai-nilai ini untuk membuat server Chef Automate Anda.

```
{
    "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. Buat AWS OpsWorks for Chef Automate server dengan menjalankan `create-server` perintah.

   - `--engine`Nilainya adalah`ChefAutomate`, `--engine-model` adalah`Single`, dan `--engine-version` adalah`12`.

- Nama server harus unik di dalam AWS akun Anda, di setiap wilayah. Nama server harus dimulai dengan huruf; kemudian huruf, angka, atau tanda hubung (-) diperbolehkan, hingga maksimum 40 karakter.
- Gunakan ARN profil instance dan ARN peran layanan yang Anda salin di Langkah 4 dan 5.
- Jenis contoh yang valid adalah`m5.large`,`r5.xlarge`, atau`r5.2xlarge`. Untuk informasi selengkapnya tentang spesifikasi jenis instans ini, lihat Jenis Instance di Panduan EC2 Pengguna Amazon.
- `--engine-attributes`Parameternya opsional; jika Anda tidak menentukan satu atau kedua nilai, proses pembuatan server menghasilkan nilai untuk Anda. Jika Anda menambahkan`--engine-attributes`, tentukan `CHEF_AUTOMATE_PIVOTAL_KEY` nilai yang Anda hasilkan di Langkah 2, a`CHEF_AUTOMATE_ADMIN_PASSWORD`, atau keduanya.

  Jika Anda tidak menetapkan nilai untuk`CHEF_AUTOMATE_ADMIN_PASSWORD`, kata sandi dibuat dan dikembalikan sebagai bagian dari `create-server` respons. Anda juga dapat mengunduh starter kit lagi di konsol, yang meregenerasi kata sandi ini. Panjang kata sandi minimal delapan karakter, dan maksimum 32. Kata sandi dapat berisi huruf, angka, dan karakter khusus (`!/@#$%^+=_`). Sandi harus berisi setidaknya satu huruf kecil, satu huruf besar, satu angka, dan satu karakter khusus.
- Sebuah key pair SSH bersifat opsional, tetapi dapat membantu Anda terhubung ke server Chef Automate Anda jika Anda perlu mengatur ulang kata sandi administrator dasbor Chef Automate. Untuk informasi selengkapnya tentang membuat key pair SSH, lihat Amazon EC2 Key Pairs di Panduan EC2 Pengguna Amazon.
- Untuk menggunakan domain kustom, tambahkan parameter berikut ke perintah Anda. Jika tidak, proses pembuatan server Chef Automate secara otomatis menghasilkan titik akhir untuk Anda. Ketiga parameter tersebut diperlukan untuk mengonfigurasi domain khusus. Untuk informasi tentang persyaratan tambahan untuk menggunakan parameter ini, lihat CreateServerdi Referensi AWS OpsWorks CM API.
  - `--custom-domain`- Sebuah endpoint publik opsional dari server, sepertihttps://`aws.my-company.com`.
  - `--custom-certificate`- Sertifikat HTTPS berformat PEM. Nilai dapat menjadi sertifikat tunggal yang ditandatangani sendiri, atau rantai sertifikat.
  - `--custom-private-key`- Kunci pribadi dalam format PEM untuk menghubungkan ke server dengan menggunakan HTTPS. Kunci privat tidak boleh dienkripsi; tidak dapat dilindungi oleh kata sandi atau frasa sandi.

- Pemeliharaan sistem mingguan diperlukan. Nilai yang valid harus ditentukan dalam format berikut:`DDD:HH:MM`. Waktu yang ditentukan adalah dalam Waktu Universal Terkoordinasi (UTC). Jika Anda tidak menentukan nilai untuk`--preferred-maintenance-window`, nilai default adalah acak, periode satu jam pada hari Selasa, Rabu, atau Jumat.
- Nilai yang valid untuk `--preferred-backup-window` harus ditentukan dalam salah satu format berikut: `HH:MM` untuk backup harian, atau `DDD:HH:MM` untuk backup mingguan. Waktu yang ditentukan adalah dalam UTC. Nilai default adalah waktu mulai acak harian. Untuk memilih keluar dari backup otomatis, tambahkan parameter `--disable-automated-backup` sebagai gantinya.
- Untuk`--security-group-ids`, masukkan satu atau lebih grup keamanan IDs, dipisahkan oleh spasi.
- Untuk`--subnet-ids`, masukkan subnet ID.

```
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
```

Berikut adalah contohnya.

```
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
```

Contoh berikut membuat server Chef Automate yang menggunakan domain kustom.

```
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
```

Contoh berikut membuat server Chef Automate yang menambahkan dua tag: `Stage: Production` dan`Department: Marketing`. Untuk informasi selengkapnya tentang menambahkan dan mengelola tag di AWS OpsWorks for Chef Automate server, lihat Bekerja dengan Tag pada AWS OpsWorks for Chef Automate Sumber Daya di panduan ini.

```
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 membutuhkan waktu sekitar 15 menit untuk membuat server baru. Jangan mengabaikan output dari `create-server` perintah atau menutup sesi shell Anda, karena output dapat berisi informasi penting yang tidak ditampilkan lagi. Untuk mendapatkan kata sandi dan starter kit dari `create-server` hasil, lanjutkan ke langkah berikutnya.

   Jika Anda menggunakan domain khusus dengan server, dalam output `create-server` perintah, salin nilai `Endpoint` atribut. Berikut adalah contohnya.

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

8. Jika Anda memilih untuk AWS OpsWorks for Chef Automate membuat kunci dan kata sandi untuk Anda, Anda dapat mengekstraknya dalam format yang dapat digunakan dari `create-server` hasil dengan menggunakan prosesor JSON seperti jq. Setelah Anda menginstal jq, Anda dapat menjalankan perintah berikut untuk mengekstrak kunci penting, kata sandi administrator dasbor Chef Automate, dan starter kit. Jika Anda tidak memberikan kunci dan kata sandi penting Anda sendiri di Langkah 4, pastikan untuk menyimpan kunci penting dan kata sandi administrator yang diekstraksi di lokasi yang nyaman namun aman.

   ```
   #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. Secara opsional, jika Anda tidak mengekstrak starter kit dari hasil `create-server` perintah, Anda dapat mengunduh starter kit baru dari halaman Properti server di AWS OpsWorks for Chef Automate konsol. Mengunduh starter kit baru akan mengatur ulang kata sandi administrator dasbor Chef Automate.

10. Jika Anda tidak menggunakan domain khusus, lanjutkan ke langkah berikutnya. Jika Anda menggunakan domain kustom dengan server, buat entri CNAME di alat manajemen DNS perusahaan Anda untuk mengarahkan domain kustom Anda ke AWS OpsWorks for Chef

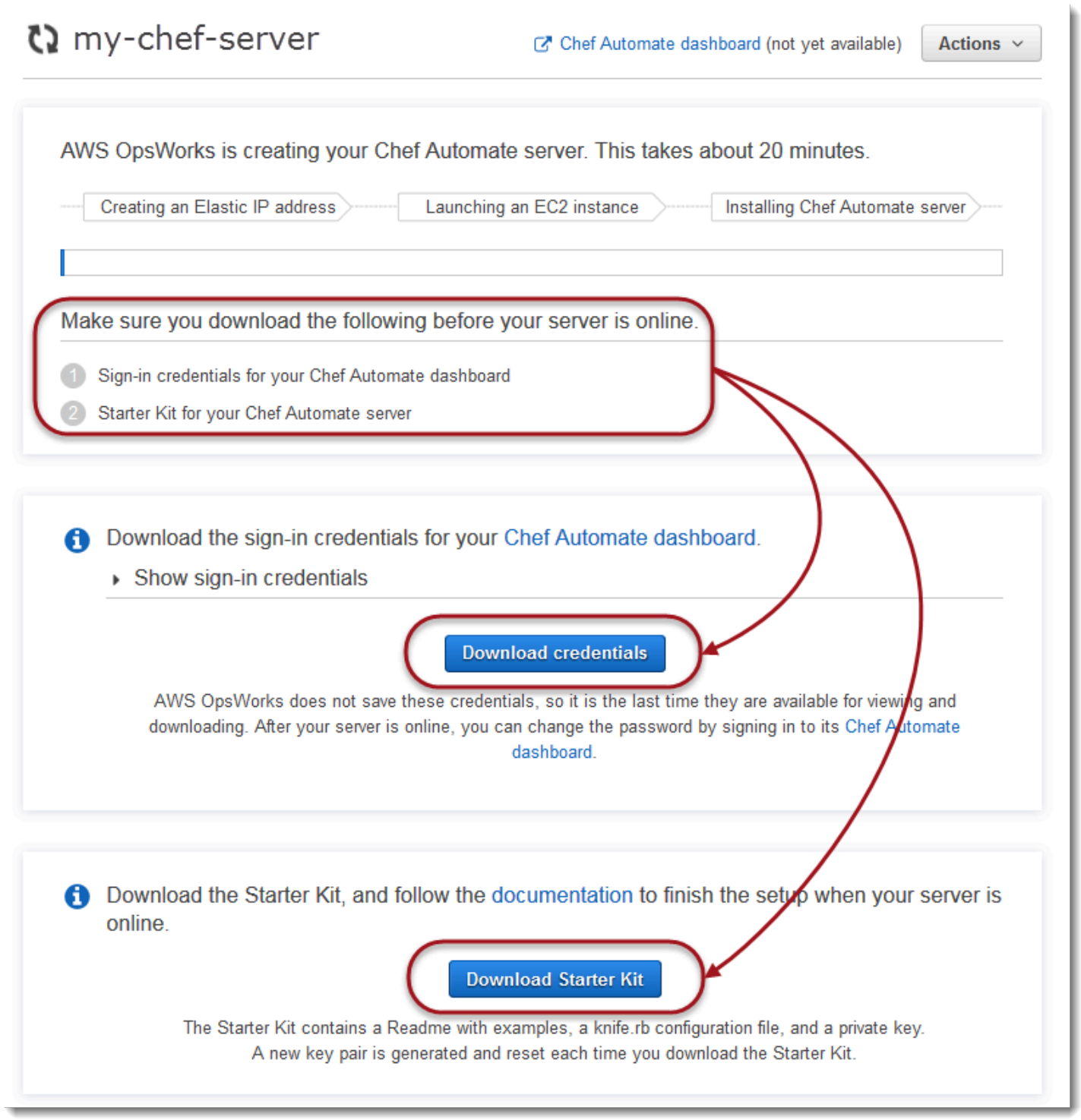
Automate titik akhir yang Anda salin di langkah 7. Anda tidak dapat menjangkau atau masuk ke server dengan domain khusus sampai Anda menyelesaikan langkah ini.

11. Ketika proses pembuatan server selesai, lanjutkan kethe section called “Selesaikan konfigurasi dan unggah buku masak”.

Konfigurasikan Server Chef Menggunakan Starter Kit

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Sementara pembuatan server Chef masih dalam proses, buka halaman Properties di AWS OpsWorks for Chef Automate konsol. Pertama kali Anda bekerja dengan server Chef baru, halaman Properties meminta Anda untuk mengunduh dua item yang diperlukan. Unduh item ini sebelum server Chef Anda online; tombol unduh tidak tersedia setelah server baru online.



- Kredensi masuk untuk server Chef. Anda akan menggunakan kredensil ini untuk masuk ke dasbor Chef Automate, tempat Anda bekerja dengan fitur premium Chef Automate, seperti alur kerja dan pemindaian kepatuhan. AWS OpsWorks tidak menyimpan kredensil ini; ini adalah terakhir kalinya

mereka tersedia untuk dilihat dan diunduh. Jika perlu, Anda dapat mengubah kata sandi yang diberikan dengan kredensil ini setelah Anda masuk.

- Kit Pemula. Starter Kit berisi file README dengan contoh, file `knife.rb` konfigurasi, dan kunci pribadi untuk pengguna utama, atau penting,. Sebuah key pair baru dihasilkan—dan kunci lama disetel ulang—setiap kali Anda mengunduh Starter Kit.

Selain kredensil yang hanya berfungsi dengan server baru, file Starter Kit .zip menyertakan contoh sederhana dari repositori Chef yang berfungsi dengan server apa pun. AWS OpsWorks for Chef Automate Di repositori Chef, Anda menyimpan buku masak, peran, file konfigurasi, dan artefak lainnya untuk mengelola node Anda dengan Chef. Kami menyarankan Anda menyimpan repositori ini dalam sistem kontrol versi, seperti Git, dan memperlakukannya sebagai kode sumber. Untuk informasi dan contoh yang menunjukkan cara menyiapkan repositori Chef yang dilacak di Git, lihat Tentang chef-repo dalam dokumentasi Chef.

Prasyarat

1. Sementara pembuatan server masih berlangsung, unduh kredensi masuk untuk server Chef, dan simpan di lokasi yang aman namun nyaman.
2. Unduh Starter Kit, dan unzip file Starter Kit .zip ke direktori ruang kerja Anda. Jangan bagikan kunci pribadi Starter Kit. Jika pengguna lain akan mengelola server Chef, tambahkan mereka sebagai administrator di dasbor Chef Automate nanti.
3. Unduh dan instal Chef Workstation (sebelumnya dikenal sebagai Chef Development Kit, atau Chef DK) di komputer yang akan Anda gunakan untuk mengelola server dan node Chef Anda. `knife`Utilitas adalah bagian dari Chef Workstation. Untuk petunjuk, lihat Menginstal Chef Workstation di situs web Chef.

Jelajahi Konten Starter Kit

Starter Kit memiliki konten berikut.

- `cookbooks/`- Direktori untuk buku masak yang Anda buat. `cookbooks/`Folder berisi `opsworks-webserver` buku masak, buku masak pembungkus yang bergantung pada buku masak dari situs web `nginx` Chef Supermarket. `Policyfile.rb`default ke supermarket Chef sebagai sumber sekunder jika dependensi buku masak tidak tersedia di direktori. `cookbooks/`
- `Policyfile.rb`- File kebijakan berbasis Ruby yang mendefinisikan buku masak, dependensi, dan atribut yang menjadi kebijakan untuk node Anda.

- `userdata.sh`dan `userdata.ps1` - Anda dapat menggunakan file data pengguna untuk mengaitkan node secara otomatis setelah meluncurkan server Chef Automate Anda. `userdata.sh`adalah untuk bootstrap node berbasis Linux, dan untuk node berbasis Windows. `userdata.ps1`
- `Berksfile`- Anda dapat menggunakan file ini jika Anda lebih suka menggunakan Berkshelf dan `berks` perintah untuk mengunggah buku masak dan dependensinya. Dalam panduan ini, kami menggunakan perintah `Policyfile.rb` dan Chef untuk mengunggah buku masak, dependensi, dan atribut.
- `README.md`, file berbasis Markdown yang menjelaskan cara menggunakan Starter Kit untuk mengatur server Chef Automate Anda untuk pertama kalinya.
- `.chef`adalah direktori tersembunyi yang berisi file konfigurasi pisau (`knife.rb`) dan file kunci otentikasi rahasia (.pem).
  - `.chef/knife.rb`- File konfigurasi pisau (`knife.rb`). `knife.rb`File dikonfigurasi sehingga operasi `knife`alat Chef berjalan melawan AWS OpsWorks for Chef Automate server.
  - `.chef/ca_certs/opsworks-cm-ca-2020-root.pem`- Kunci pribadi SSL yang ditandatangani oleh otoritas sertifikasi (CA) yang disediakan oleh. AWS OpsWorks Kunci ini memungkinkan server untuk mengidentifikasi dirinya ke agen klien Chef Infra pada node yang dikelola server Anda.

Siapkan Repositori Chef Anda

Sebuah repositori Chef berisi beberapa direktori. Setiap direktori di Starter Kit berisi file README yang menjelaskan tujuan direktori, dan cara menggunakannya untuk mengelola sistem Anda dengan Chef. Ada dua cara untuk menginstal buku masak di server Chef Anda: menjalankan `knife` perintah, atau menjalankan perintah Chef untuk mengunggah file kebijakan (`Policyfile.rb`) ke server Anda yang mengunduh dan menginstal buku masak tertentu. Panduan ini menggunakan perintah Chef dan `Policyfile.rb` menginstal buku masak di server Anda.

1. Buat direktori di komputer lokal Anda untuk menyimpan buku masak, seperti`chef-repo`. Setelah Anda menambahkan buku masak, peran, dan file lain ke repositori ini, kami sarankan Anda mengunggah atau menyimpannya dalam sistem yang aman dan berversi, seperti, CodeCommit Git, atau Amazon S3.
2. Di `chef-repo` direktori, buat direktori berikut:
   - `cookbooks/`- Toko buku masak.
   - `roles/`- Menyimpan peran dalam `.rb` atau `.json` format.

- `environments/`- Menyimpan lingkungan dalam `.rb` atau `.json` format.

Gunakan PolicyFile.rb untuk Mendapatkan Buku Masak dari Sumber Jarak Jauh

Di bagian ini, edit `Policyfile.rb` untuk menentukan buku masak, lalu jalankan perintah Chef untuk mengunggah file ke server Anda dan menginstal buku masak.

1. Lihat `Policyfile.rb` di Starter Kit Anda. Secara default, `Policyfile.rb` sertakan buku masak `opsworks-webserver` pembungkus, yang tergantung pada `nginx`buku masak yang tersedia di situs web Chef Supermarket. `nginx`Buku masak menginstal dan mengkonfigurasi server web pada node terkelola. `chef-client`Buku masak yang diperlukan, yang menginstal agen klien Chef Infra pada node terkelola, juga ditentukan.

   `Policyfile.rb`juga menunjuk ke buku masak Chef Audit opsional, yang dapat Anda gunakan untuk mengatur pemindaian kepatuhan pada node. Untuk informasi selengkapnya tentang menyiapkan pemindaian kepatuhan dan mendapatkan hasil kepatuhan untuk node terkelola, lihatPemindaian Kepatuhan di AWS OpsWorks for Chef Automate. Jika Anda tidak ingin mengonfigurasi pemindaian kepatuhan dan audit sekarang, hapus `'audit'` dari `run_list` bagian, dan jangan tentukan atribut `audit` buku masak di akhir file.

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

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


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

name 'opsworks-demo-webserver'

# The cookbooks directory is the preferred source for external cookbooks

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

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

end
# Alternative source
```

```
default_source :supermarket


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

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

# Specify a custom source for a single cookbook:

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


# Policyfile defined attributes

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

Berikut ini adalah contoh `Policyfile.rb` tanpa `audit` buku masak dan atribut, jika Anda ingin mengkonfigurasi hanya server `nginx` web untuk saat ini.

```
# 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'
```

Jika Anda membuat perubahan`Policyfile.rb`, pastikan untuk menyimpan file.

2. Unduh dan instal buku masak yang ditentukan dalam`Policyfile.rb`.

```
chef install
```

Semua buku masak diberi versi dalam file buku masak. `metadata.rb` Setiap kali Anda mengganti buku masak, Anda harus menaikkan versi buku masak yang ada di dalamnya. `metadata.rb`

3. Jika Anda memilih untuk mengonfigurasi pemindaian kepatuhan, dan menyimpan informasi `audit` buku masak di file kebijakan, dorong kebijakan `opsworks-demo` ke server Anda.

```
chef push opsworks-demo
```

4. Jika Anda menyelesaikan langkah 3, verifikasi penginstalan kebijakan Anda. Jalankan perintah berikut.

```
chef show-policy
```

Hasilnya harus menyerupai yang berikut:

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

5. Anda sekarang siap untuk menambahkan atau mem-bootstrap node ke server Chef Automate Anda. Anda dapat mengotomatiskan asosiasi node dengan mengikuti langkah-langkah[Tambahkan node secara otomatis di AWS OpsWorks for Chef Automate](), atau menambahkan node satu per satu dengan mengikuti langkah-langkah dalam[Tambahkan node satu per satu]().

(Alternatif) Gunakan Berkshelf untuk Mendapatkan Buku Masak dari Sumber Jarak Jauh

Berkshelf adalah alat untuk mengelola buku masak dan dependensinya. Jika Anda lebih suka menggunakan Berkshelf daripada `Policyfile.rb` menginstal buku masak ke penyimpanan lokal, gunakan prosedur di bagian ini alih-alih bagian sebelumnya. Anda dapat menentukan buku masak dan versi mana yang akan digunakan dengan server Chef Anda dan mengunggahnya. Starter Kit berisi file bernama `Berksfile` yang dapat Anda gunakan untuk membuat daftar buku masak Anda.

1. Untuk memulai, tambahkan `chef-client` buku masak ke Berksfile yang disertakan. `chef-client`Buku masak mengonfigurasi perangkat lunak agen klien Chef Infra pada setiap node yang Anda sambungkan ke server Chef Automate Anda. Untuk mempelajari lebih lanjut tentang buku masak ini, lihat Buku Masak Klien Chef di Supermarket Chef.
2. Menggunakan editor teks, tambahkan buku masak lain ke Berksfile Anda yang menginstal aplikasi server web; misalnya, `apache2` buku masak, yang menginstal server web Apache. Berksfile Anda harus menyerupai yang berikut ini.

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

3. Unduh dan instal buku masak di komputer lokal Anda.

   ```
   berks install
   ```

4. Unggah buku masak ke server Chef.

   Di Linux, jalankan yang berikut ini.

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

   Di Windows, jalankan perintah Chef Workstation berikut dalam satu PowerShell sesi. Sebelum Anda menjalankan perintah, pastikan untuk mengatur kebijakan eksekusi PowerShell ke`RemoteSigned`. Tambahkan `chef shell-init` untuk membuat perintah utilitas Chef Workstation tersedia untuk 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. Verifikasi pemasangan buku masak dengan menampilkan daftar buku masak yang saat ini tersedia di server Chef Automate. Anda dapat melakukan ini dengan menjalankan `knife` perintah berikut.

   Anda siap menambahkan node untuk dikelola dengan AWS OpsWorks for Chef Automate server.

   ```
   knife cookbook list
   ```

(Opsional) Konfigurasi **`knife`** untuk Bekerja dengan Domain Kustom

Jika server Chef Automate Anda menggunakan domain khusus, Anda mungkin perlu menambahkan sertifikat PEM dari root CA yang menandatangani rantai sertifikat server Anda, atau sertifikat PEM server Anda jika sertifikat ditandatangani sendiri. `ca_certs`adalah subdirektori `chef/` yang berisi otoritas sertifikat (CAs) yang dipercaya oleh `knife` utilitas Chef.

Anda dapat melewati bagian ini jika Anda tidak menggunakan domain kustom, atau jika sertifikat kustom Anda ditandatangani oleh CA root yang dipercaya oleh sistem operasi Anda. Jika tidak, konfigurasikan `knife` untuk mempercayai sertifikat SSL server Chef Automate Anda, seperti yang dijelaskan dalam langkah-langkah berikut.

1. Jalankan perintah berikut.

   ```
   knife ssl check
   ```

   Jika hasilnya mirip dengan yang berikut ini, lewati sisa prosedur ini, dan lanjutkan keTambahkan Node untuk Server Chef untuk Dikelola.

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

   Jika Anda mendapatkan pesan kesalahan yang mirip dengan berikut ini, lanjutkan ke langkah berikutnya.

   ```
   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. Jalankan `knife ssl fetch` untuk mempercayai sertifikat AWS OpsWorks for Chef Automate server Anda. Atau, Anda dapat secara manual menyalin sertifikat berformat CA PEM root dari server Anda ke direktori yang merupakan nilai `trusted_certs_dir` dalam output dari. `knife ssl check` Secara default, direktori ini ada `.chef/ca_certs/` di Starter Kit. Output Anda harus menyerupai yang berikut:

```
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. Jalankan lagi `knife ssl check`. Output Anda harus menyerupai yang berikut:

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

Anda siap digunakan `knife` dengan server Chef Automate Anda.

Tambahkan Node untuk Server Chef untuk Dikelola

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan,

Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

`chef-client`Agen menjalankan resep Chef di komputer fisik atau virtual, yang disebut node, yang terkait dengan server. Anda dapat menghubungkan komputer atau instance lokal ke server Chef untuk dikelola, asalkan node menjalankan sistem operasi yang didukung. Mendaftarkan node dengan server Chef menginstal perangkat lunak `chef-client` agen pada node tersebut.

Anda dapat menggunakan metode berikut untuk menambahkan node:

- Tambahkan catatan satu per satu dengan menjalankan `knife` perintah yang menambahkan, atau bootstrap, sebuah EC2 instance sehingga server Chef dapat mengelolanya. Untuk mengetahui informasi selengkapnya, lihat Tambahkan node satu per satu.
- Tambahkan node secara otomatis dengan menggunakan skrip untuk melakukan asosiasi node tanpa pengawasan dengan server Chef. Kode dalam Starter Kit menunjukkan cara menambahkan node secara otomatis menggunakan metode tanpa pengawasan. Untuk informasi selengkapnya, lihat Tambahkan node secara otomatis di AWS OpsWorks for Chef Automate.

Topik

- Tambahkan node satu per satu
- Tambahkan node secara otomatis di AWS OpsWorks for Chef Automate

Tambahkan node satu per satu

Important

AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara menjalankan `knife` perintah yang menambahkan, atau bootstrap, sebuah EC2 instance sehingga server Chef dapat mengelolanya.

Versi minimum yang didukung `chef-client` pada node yang terkait dengan AWS OpsWorks for Chef Automate server adalah 13. x. Kami merekomendasikan menjalankan `chef-client` versi terbaru dan stabil.

Topik

- (Opsional) Tentukan URL CA Root Server Chef Automate Anda
- Sistem Operasi yang Didukung
- Tambahkan Node dengan Pisau

(Opsional) Tentukan URL CA Root Server Chef Automate Anda

Jika server Anda menggunakan domain dan sertifikat khusus, Anda mungkin perlu mengedit `ROOT_CA_URL` variabel dalam skrip data pengguna dengan URL publik yang dapat Anda gunakan untuk mendapatkan sertifikat berformat CA PEM root dari server Anda. AWS CLI Perintah berikut mengunggah CA root Anda ke bucket Amazon S3, dan buat URL presigned yang dapat Anda gunakan selama satu jam.

1. Unggah sertifikat berformat CA PEM root ke S3.

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

2. Buat URL presigned yang dapat Anda gunakan selama satu jam (3600 detik, dalam contoh ini) untuk mengunduh root CA.

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

3. Edit variabel `ROOT_CA_URL` dalam skrip data pengguna dengan nilai URL yang telah ditandatangani sebelumnya.

Sistem Operasi yang Didukung

Untuk daftar sistem operasi yang didukung untuk node saat ini, lihat situs web Chef.

Tambahkan Node dengan Pisau

`knife-ec2`Plug-in disertakan dengan Chef Workstation. Jika Anda lebih akrab dengan`knife-ec2`, Anda dapat menggunakannya alih-alih `knife bootstrap` untuk penyediaan dan bootstrap baru EC2instances. Jika tidak, luncurkan EC2 instance baru, lalu ikuti langkah-langkah di bagian ini.

Untuk menambahkan node untuk dikelola

1. Jalankan perintah `knife bootstrap` berikut. Perintah ini mem-boot EC2 instance ke node yang akan dikelola server Chef Anda. Perhatikan bahwa Anda menginstruksikan server Chef untuk menjalankan resep dari `nginx` buku masak yang Anda instal. the section called “Gunakan PolicyFile.rb untuk Mendapatkan Buku Masak dari Sumber Jarak Jauh” Untuk informasi selengkapnya tentang menambahkan node dengan menjalankan `knife bootstrap` perintah, lihat Bootstrap Node dalam dokumentasi Chef.

   Tabel berikut menunjukkan nama pengguna yang valid untuk sistem operasi node dalam `knife` perintah dalam langkah ini. Jika tidak `root` atau `ec2-user` berfungsi, tanyakan kepada penyedia AMI Anda. Untuk informasi selengkapnya tentang menghubungkan ke instans berbasis Linux, lihat Menghubungkan ke Instans Linux Anda Menggunakan SSH di dokumentasi AWS.

   Nilai yang valid untuk nama pengguna di sistem operasi node

Sistem Operasi	Nama Pengguna yang Valid
Amazon Linux	`ec2-user`
Perusahaan Topi Merah Linux 5	`root` atau `ec2-user`
Ubuntu	`ubuntu`
Fedora	`fedora` atau `ec2-user`
SUSE Linux	`root` atau `ec2-user`

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

2. Verifikasi bahwa node baru ditambahkan dengan menjalankan perintah berikut, ganti *INSTANCE_NAME* dengan nama instance yang baru saja Anda tambahkan.

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

Tambahkan node secara otomatis di AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Topik ini menjelaskan cara menambahkan node Amazon Elastic Compute Cloud (Amazon EC2) ke server Chef Anda secara otomatis. Kode dalam Starter Kit menunjukkan cara menambahkan node secara otomatis menggunakan metode tanpa pengawasan. Metode yang direkomendasikan untuk asosiasi node baru tanpa pengawasan (atau otomatis) adalah dengan mengkonfigurasi Chef Client Cookbook. Anda dapat menggunakan `userdata` skrip di Starter Kit, dan mengubah `run_list` bagian `userdata` skrip, atau `Policyfile.rb` dengan buku masak yang ingin Anda terapkan ke node Anda. Sebelum Anda menjalankan `chef-client` agen, instal buku masak Chef Client di server Chef Anda, lalu instal `chef-client` agen dalam mode layanan dengan, misalnya, peran HTTPD, seperti yang ditunjukkan pada perintah contoh berikut.

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

Untuk berkomunikasi dengan server Chef, perangkat lunak `chef-client` agen harus memiliki akses ke kunci publik node klien. Anda dapat membuat key pair public-private di Amazon EC2, lalu meneruskan kunci publik ke panggilan AWS OpsWorks `associate-node` API dengan nama node. Skrip yang disertakan dalam Starter Kit mengumpulkan nama organisasi, nama server, dan titik akhir server untuk Anda. Ini memastikan bahwa node dikaitkan dengan server Chef, dan perangkat lunak `chef-client` agen yang berjalan pada node dapat berkomunikasi dengan server setelah mencocokkan kunci pribadi.

Versi minimum yang didukung `chef-client` pada node yang terkait dengan AWS OpsWorks for Chef Automate server adalah 13. x. Kami merekomendasikan menjalankan `chef-client` versi terbaru dan stabil.

Untuk informasi tentang cara memisahkan node, lihat Memutuskan Node dari Server AWS OpsWorks for Chef Automate di panduan ini, dan `disassociate-node`di dokumentasi AWS OpsWorks for Chef Automate API.

Topik

- Sistem Operasi yang Didukung
- Langkah 1: Buat Peran IAM untuk Digunakan sebagai Profil Instance Anda
- Langkah 2: Instal Buku Masak Klien Chef
- Langkah 3: Buat Instans dengan Menggunakan Skrip Asosiasi Tanpa Pengawasan
- Metode Lain untuk Mengotomatisasi Berulang Berulang chef-client
- Topik Terkait

Sistem Operasi yang Didukung

Untuk daftar sistem operasi yang didukung untuk node saat ini, lihat situs web Chef.

Langkah 1: Buat Peran IAM untuk Digunakan sebagai Profil Instance Anda

Buat peran AWS Identity and Access Management (IAM) untuk digunakan sebagai profil EC2 instans Anda, dan lampirkan kebijakan berikut ke peran IAM. Kebijakan ini memungkinkan AWS OpsWorks for Chef Automate (`opsworks-cm`) API untuk berkomunikasi dengan EC2 instance selama pendaftaran node. Untuk informasi selengkapnya tentang profil instans, lihat Menggunakan Profil Instance di EC2 dokumentasi Amazon. Untuk informasi tentang cara membuat peran IAM, lihat Membuat Peran IAM di Konsol di dokumentasi Amazon EC2 .

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

Langkah 2: Instal Buku Masak Klien Chef

Jika Anda belum melakukannya, ikuti langkah-langkah (Alternatif) Gunakan Berkshelf untuk Mendapatkan Buku Masak dari Sumber Jarak Jauh untuk memastikan bahwa Berksfile atau `Policyfile.rb` file Anda mereferensikan buku masak Chef Client dan menginstal buku masak.

Langkah 3: Buat Instans dengan Menggunakan Skrip Asosiasi Tanpa Pengawasan

1. Untuk membuat EC2 instance, Anda dapat menyalin `userdata` skrip dari Starter Kit ke `userdata` bagian instruksi EC2 instance, konfigurasi peluncuran grup Amazon EC2 Auto Scaling, atau templat. AWS CloudFormation Untuk informasi selengkapnya tentang menambahkan skrip ke data pengguna, lihat Menjalankan Perintah pada Instans Linux Anda saat Peluncuran di EC2 dokumentasi Amazon.

   Skrip ini menjalankan `associate-node`perintah `opsworks-cm` API untuk mengaitkan node baru dengan server Chef Anda.

   Secara default, nama node terdaftar baru adalah ID instance, tetapi Anda dapat mengubah nama dengan memodifikasi nilai `NODE_NAME` variabel dalam `userdata` skrip. Karena mengubah nama organisasi di UI konsol Chef saat ini tidak dimungkinkan, biarkan `CHEF_AUTOMATE_ORGANIZATION` disetel ke`default`.
2. Ikuti prosedur dalam Meluncurkan Instance dalam EC2 dokumentasi, dengan modifikasi di sini. Di wizard peluncuran EC2 instance, pilih AMI Amazon Linux.
3. Pada halaman Konfigurasi Detail Instance, pilih peran yang Anda buatLangkah 1: Buat Peran IAM untuk Digunakan sebagai Profil Instance Anda, sebagai peran IAM Anda.
4. Di area Detail Lanjutan, unggah `userdata.sh` skrip yang Anda buat sebelumnya dalam prosedur ini.
5. Tidak diperlukan perubahan pada halaman Add Storage. Lanjutkan ke Tambahkan Tag.
6. Pada halaman Configure Security Group, pilih Add Rule, lalu pilih tipe HTTP untuk membuka nomor port 443 dan 80 untuk server web Apache dalam contoh ini.
7. Pilih Tinjau dan Luncurkan, lalu pilih Luncurkan. Ketika node baru Anda dimulai, itu menerapkan konfigurasi yang ditentukan oleh resep yang telah Anda tentukan dalam `RUN_LIST` parameter.
8. Opsional: Jika Anda telah menambahkan `nginx` buku masak ke daftar run Anda, ketika Anda membuka halaman web yang ditautkan ke DNS publik dari node baru Anda, Anda akan melihat situs web yang di-host oleh server web nginx Anda.

Metode Lain untuk Mengotomatisasi Berulang Berulang **`chef-client`**

Meskipun lebih sulit dicapai, dan tidak disarankan, Anda dapat menjalankan skrip dalam topik ini semata-mata sebagai bagian dari data pengguna instance mandiri, menggunakan AWS CloudFormation templat untuk menambahkannya ke data pengguna instance baru, mengonfigurasi `cron` pekerjaan untuk menjalankan skrip secara teratur, atau menjalankan `chef-client` dalam layanan. Namun, kami merekomendasikan metode Chef Client Cookbook karena ada beberapa kelemahan dengan teknik otomatisasi lainnya.

Untuk daftar lengkap parameter yang dapat Anda berikan`chef-client`, lihat dokumentasi Chef.

Topik Terkait

Posting AWS blog berikut menawarkan informasi lebih lanjut tentang menghubungkan node secara otomatis dengan server Chef Automate Anda, dengan menggunakan grup Auto Scaling, atau dalam beberapa akun.

- Menggunakan AWS OpsWorks for Chef Automate untuk Mengelola EC2 Instans dengan Auto Scaling
- OpsWorks untuk Chef Automate - Secara Otomatis Bootstrapping Node di Akun yang Berbeda

Masuk ke dasbor Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Setelah Anda mengunduh kredensi masuk dari halaman Properti server Chef, dan server sedang online, masuk ke dasbor Chef Automate. Dalam panduan ini, kami menginstruksikan Anda untuk mengunggah buku masak terlebih dahulu dan menambahkan setidaknya satu node untuk dikelola. Ini memungkinkan Anda untuk melihat informasi tentang buku masak dan node di dasbor.

Ketika Anda mencoba untuk terhubung ke halaman web dasbor, peringatan sertifikat muncul di browser Anda sampai Anda menginstal sertifikat SSL AWS OpsWorks khusus yang ditandatangani

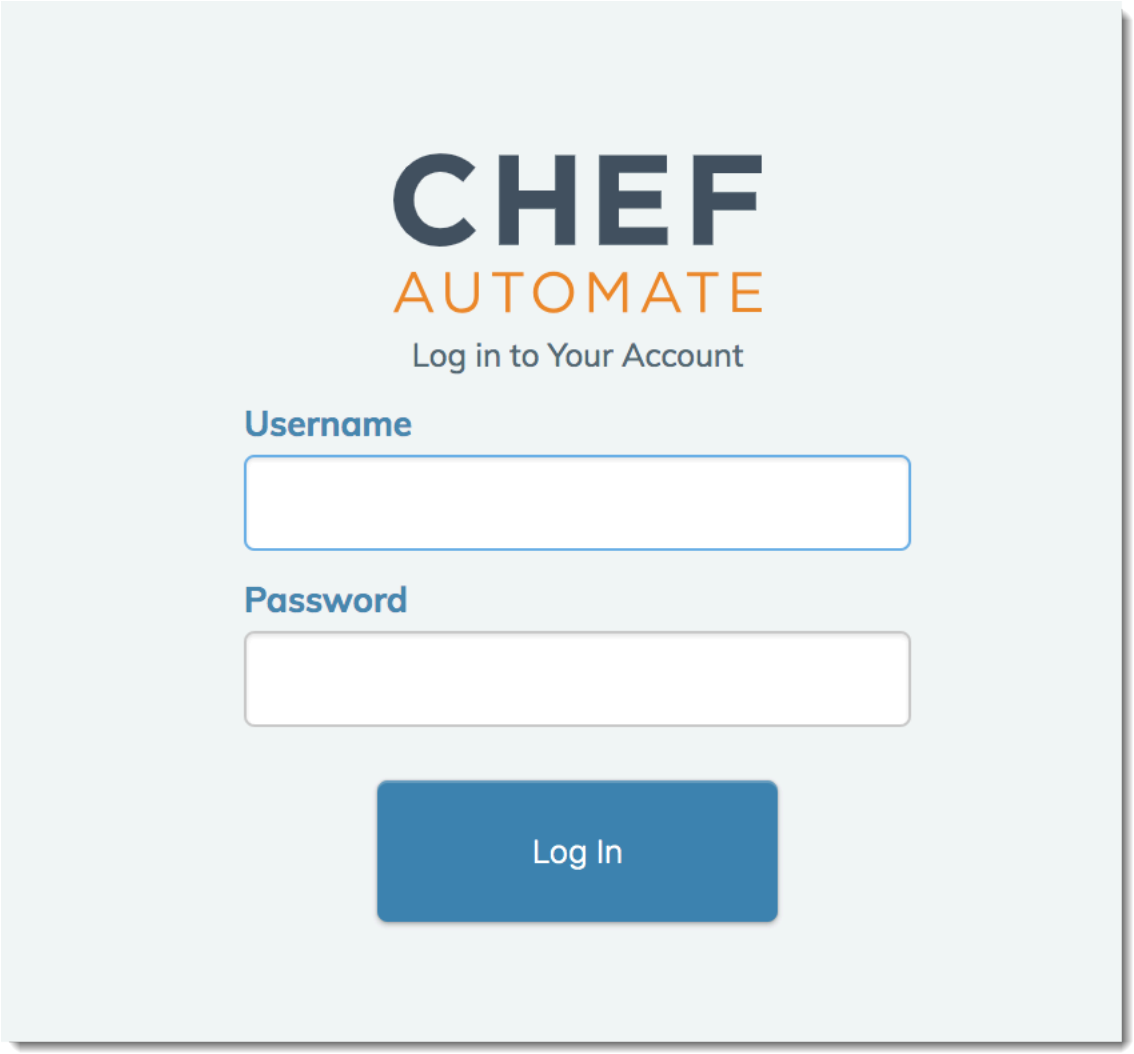
CA pada komputer klien yang Anda gunakan untuk mengelola server Chef Anda. Jika Anda memilih untuk tidak melihat peringatan sebelum melanjutkan ke halaman web dasbor, instal sertifikat SSL sebelum Anda masuk.

Setelah Anda menginstal sertifikat SSL sisi klien, Anda dapat masuk ke dasbor Chef Automate tanpa melihat pesan peringatan.

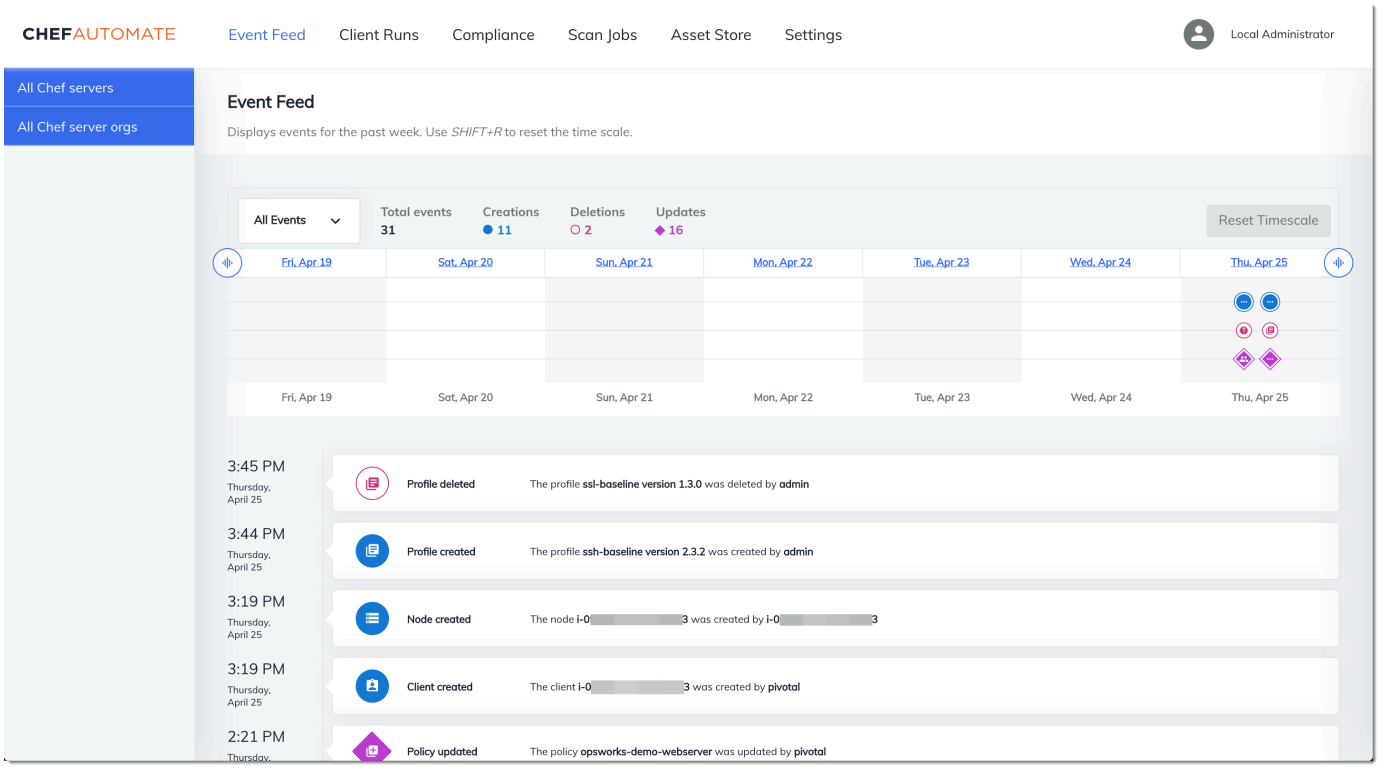
> **Note**
>
> Pengguna Google Chrome di sistem operasi Ubuntu dan Linux Mint dapat mengalami kesulitan masuk. Kami menyarankan Anda menggunakan Mozilla Firefox atau browser lain untuk masuk dan menggunakan dasbor Chef Automate pada sistem operasi tersebut. Tidak ada masalah yang ditemukan menggunakan Google Chrome di Windows atau macOS.

Untuk masuk ke dasbor Chef Automate

1. Buka zip dan buka kredenal Chef Automate yang Anda unduh. Prasyarat Anda akan memerlukan kredensi ini untuk masuk.
2. Buka halaman Properti untuk server Chef Anda.
3. Di kanan atas halaman Properties, pilih Open Chef Automate dashboard.
4. Masuk menggunakan kredensi dari Langkah 1.



5. Di dasbor Chef Automate, Anda dapat melihat informasi terperinci tentang node yang telah Anda bootstrap, kemajuan dan acara menjalankan buku masak, tingkat kepatuhan node, dan banyak lagi. Untuk informasi selengkapnya tentang fitur dasbor Chef Automate dan cara menggunakannya, lihat Dokumentasi Otomatis Chef.



Note

Untuk informasi tentang cara mengubah kata sandi yang Anda gunakan untuk masuk ke dasbor Chef Automate, lihatAtur Ulang Kredensi Dasbor Otomatis Chef.

Membuat AWS OpsWorks for Chef Automate Server dengan menggunakan AWS CloudFormation

Important

AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks for Chef Automate memungkinkan Anda menjalankan server Chef Automate di AWS. Anda dapat menyediakan server Chef Automate dalam waktu sekitar 15 menit.

Mulai 3 Mei 2021, AWS OpsWorks for Chef Automate menyimpan beberapa atribut server Chef Automate di AWS Secrets Manager. Untuk informasi selengkapnya, lihat Integrasi dengan AWS Secrets Manager.

Panduan berikut membantu Anda membuat server AWS OpsWorks for Chef Automate dengan membuat tumpukan. AWS CloudFormation

Topik

- Prasyarat
- Buat Server Otomatis Koki di AWS CloudFormation

Prasyarat

Sebelum Anda membuat server Chef Automate baru, buat sumber daya di luar AWS OpsWorks for Chef Automate yang Anda perlukan untuk mengakses dan mengelola server Chef Anda. Untuk informasi selengkapnya, lihat Prasyarat di bagian Memulai panduan ini.

Tinjau bagian OpsWorks -CM dari Referensi Template Panduan AWS CloudFormation Pengguna untuk mempelajari nilai yang didukung dan yang diperlukan dalam AWS CloudFormation templat yang Anda gunakan untuk membuat server.

Jika Anda membuat server yang menggunakan domain khusus, Anda memerlukan domain kustom, sertifikat, dan kunci pribadi. Anda harus menentukan nilai untuk ketiga parameter ini di AWS CloudFormation template Anda. Untuk informasi selengkapnya tentang persyaratan untuk`CustomDomain`,`CustomCertificate`, dan `CustomPrivateKey` parameter, lihat CreateServerdi Referensi AWS OpsWorks CM API.

Buat nilai kata sandi untuk atribut `CHEF_AUTOMATE_ADMIN_PASSWORD` engine. Panjang kata sandi minimal delapan karakter, dan maksimum 32. Kata sandi dapat berisi huruf, angka, dan karakter khusus(`!/@#$%^+=_`). Sandi harus berisi setidaknya satu huruf kecil, satu huruf besar, satu angka, dan satu karakter khusus. Anda menentukan kata sandi ini di AWS CloudFormation template Anda, atau sebagai nilai `CHEF_AUTOMATE_ADMIN_PASSWORD` parameter saat Anda membuat tumpukan Anda.

Buat key pair RSA berenkode base64 sebelum Anda mulai membuat server Chef Automate. AWS CloudFormation Kunci publik pasangan adalah nilai`CHEF_AUTOMATE_PIVOTAL_KEY`, khusus

Chef EngineAttributesdari CreateServerAPI. Kunci ini disediakan sebagai nilai Parameter di AWS CloudFormation konsol, atau dalam create-stack perintah di AWS CLI. Untuk menghasilkan kunci ini, kami menyarankan metode berikut.

- Pada komputer berbasis Linux, Anda dapat menghasilkan kunci ini dengan menjalankan perintah OpenSSL berikut.

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

  Kemudian, ekspor bagian kunci publik RSA dari pasangan ke file. Kunci publik menjadi nilai`CHEF_AUTOMATE_PIVOTAL_KEY`.

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

- Pada komputer berbasis Windows, Anda dapat menggunakan TTYgen utilitas Pu untuk menghasilkan key pair RSA berenkode base64. Untuk informasi lebih lanjut, lihat Pu TTYgen - Key Generator untuk Putty di Windows di SSH.com.

Buat Server Otomatis Koki di AWS CloudFormation

Bagian ini menjelaskan cara menggunakan AWS CloudFormation template untuk membangun tumpukan yang membuat AWS OpsWorks for Chef Automate server. Anda dapat melakukan ini dengan menggunakan AWS CloudFormation konsol atau AWS CLI. Contoh AWS CloudFormation template tersedia untuk Anda gunakan untuk membangun tumpukan AWS OpsWorks for Chef Automate server. Pastikan untuk memperbarui template contoh dengan nama server Anda sendiri, peran IAM, profil instance, deskripsi server, jumlah retensi cadangan, opsi pemeliharaan, dan tag opsional. Jika server Anda akan menggunakan domain kustom, Anda harus menentukan nilai untuk`CustomDomain`,`CustomCertificate`, dan `CustomPrivateKey` parameter dalam AWS CloudFormation template Anda. Anda dapat menentukan atribut `CHEF_AUTOMATE_ADMIN_PASSWORD` dan `CHEF_AUTOMATE_PIVOTAL_KEY` engine dan nilainya di AWS CloudFormation template, atau hanya memberikan atribut, lalu menentukan nilai untuk atribut di wizard atau create-stack perintah AWS CloudFormation Create Stack. Untuk informasi selengkapnya tentang atribut ini, lihat the section called “Buat server Chef Automate di AWS Management Console” di bagian Memulai panduan ini.

Topik

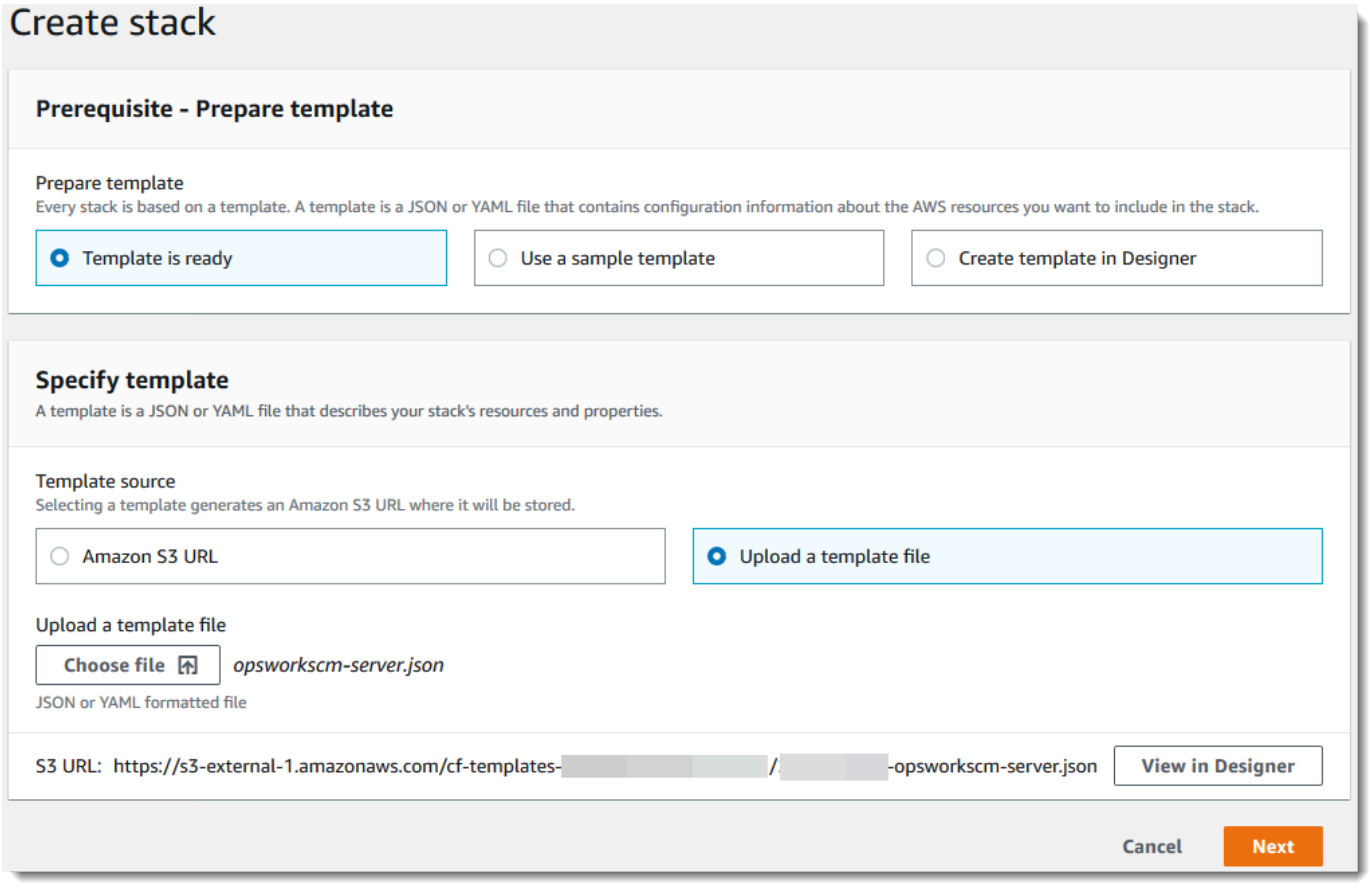
- Buat Server Otomatis Chef dengan menggunakan AWS CloudFormation (Konsol)

- Buat Server Otomatis Koki dengan menggunakan AWS CloudFormation (CLI)

Buat Server Otomatis Chef dengan menggunakan AWS CloudFormation (Konsol)

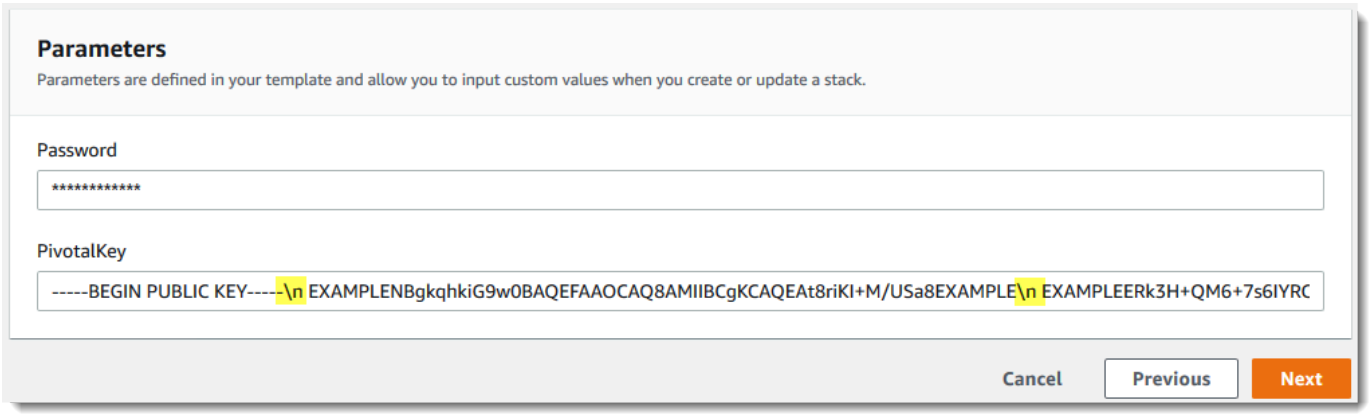
1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di AWS CloudFormation halaman beranda, pilih Buat tumpukan.
3. Dalam Prasyarat - Siapkan template, jika Anda menggunakan contoh AWS CloudFormation template, pilih Template siap.
4. Di Tentukan template, pilih sumber template Anda. Untuk panduan ini, pilih Unggah file templat, dan unggah AWS CloudFormation templat yang membuat server Chef Automate. Jelajahi file template Anda, lalu pilih Berikutnya.

   AWS CloudFormation Template dapat dalam format YAMAL atau JSON. Contoh AWS CloudFormation template tersedia untuk Anda gunakan; pastikan untuk mengganti nilai contoh dengan milik Anda sendiri. Anda dapat menggunakan desainer AWS CloudFormation template untuk membangun template baru atau memvalidasi yang sudah ada. Untuk informasi selengkapnya tentang cara melakukannya, lihat Ikhtisar Antarmuka AWS CloudFormation Desainer di Panduan AWS CloudFormation Pengguna.



5. Pada halaman Tentukan detail tumpukan, masukkan nama untuk tumpukan Anda. Ini tidak akan sama dengan nama server Anda, itu hanya nama tumpukan. Di area Parameter, tempel nilai yang Anda buatthe section called “Prasyarat”. Masukkan kata sandi di Kata Sandi.

   Rekatkan isi file kunci RSA di PivotalKey. Di AWS CloudFormation konsol, Anda harus menambahkan karakter baris baru (**\n**) di akhir setiap baris dari nilai kunci penting, seperti yang ditunjukkan pada gambar berikut. Pilih Berikutnya.



6. Pada halaman Configure stack options, Anda dapat menambahkan tag ke server yang Anda buat dengan stack, dan memilih peran IAM untuk membuat resource jika Anda belum menentukan peran IAM untuk digunakan dalam template Anda. Setelah selesai menentukan opsi, pilih Berikutnya. Untuk informasi selengkapnya tentang opsi lanjutan seperti pemicu rollback, lihat Mengatur Opsi AWS CloudFormation Tumpukan di Panduan Pengguna.AWS CloudFormation
7. Pada halaman Tinjau, tinjau pilihan Anda. Saat Anda siap membuat tumpukan server, pilih Buat tumpukan.

   Saat Anda menunggu AWS CloudFormation untuk membuat tumpukan, lihat status pembuatan tumpukan. Jika pembuatan tumpukan gagal, tinjau pesan kesalahan yang ditampilkan di konsol untuk membantu Anda menyelesaikan masalah. Untuk informasi selengkapnya tentang pemecahan masalah kesalahan di AWS CloudFormation tumpukan, lihat Memecahkan Masalah Kesalahan di Panduan Pengguna.AWS CloudFormation

   Ketika pembuatan server selesai, server Chef Automate Anda tersedia di AWS OpsWorks for Chef Automate halaman beranda, dengan status online. Hasilkan Starter Kit baru dan kredenal dasbor Chef Automate dari halaman Properti server. Setelah server online, dasbor Chef Automate tersedia di domain server, di URL dalam format berikut:`https://`*`your_server_name`*`-`*`randomID`*`.`*`region`*`.opsworks-cm.io`.

   > **Note**
   >
   > Jika Anda menetapkan domain kustom, sertifikat, dan kunci pribadi untuk server Anda, buat entri CNAME di alat manajemen DNS perusahaan Anda yang memetakan domain kustom Anda ke titik akhir yang dibuat AWS OpsWorks for Chef Automate secara otomatis untuk server. Anda tidak dapat mengelola server atau terhubung ke dasbor Chef Automate untuk server hingga Anda memetakan titik akhir yang dihasilkan ke nilai domain kustom Anda.
   > Untuk mendapatkan nilai endpoint yang dihasilkan, jalankan AWS CLI perintah berikut setelah server Anda online:
   >
   > ```
   > aws opsworks describe-servers --server-name server_name
   > ```

Buat Server Otomatis Koki dengan menggunakan AWS CloudFormation (CLI)

Jika komputer lokal Anda belum menjalankan AWS CLI, unduh dan instal AWS CLI dengan mengikuti petunjuk penginstalan di Panduan Pengguna Antarmuka Baris Perintah AWS. Bagian ini tidak menjelaskan semua parameter yang dapat Anda gunakan dengan create-stack perintah. Untuk informasi selengkapnya tentang create-stack parameter, lihat create-stackdi AWS CLI Referensi.

1. Pastikan untuk menyelesaikan Prasyarat untuk membuat AWS OpsWorks for Chef Automate server.
2. Buat peran layanan dan profil instance.

   Temukan dan salin ARNs peran layanan di akun Anda.

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

   Dalam hasil `list-roles` perintah, cari peran layanan dan entri profil instance yang menyerupai berikut ini. Catat peran layanan dan profil instance, dan tambahkan ke AWS CloudFormation template yang Anda gunakan untuk membuat tumpukan server Anda. ARNs

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

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

3. Buat AWS OpsWorks for Chef Automate server dengan menjalankan create-stack perintah lagi.

   - Ganti *`stack_name`* dengan nama tumpukan Anda. Ini adalah nama AWS CloudFormation tumpukan, bukan server Chef Automate Anda. Nama server Chef Automate adalah nilai `ServerName` dalam AWS CloudFormation template Anda.
   - Ganti *`template`* dengan path ke file template Anda, dan ekstensi *`yaml or json`* dengan `.yaml` atau yang `.json` sesuai.
   - Nilai untuk `--parameters` sesuai dengan EngineAttributesdari CreateServerAPI. Untuk Chef, atribut mesin yang disediakan pengguna untuk membuat server adalah`CHEF_AUTOMATE_PIVOTAL_KEY`, kunci publik RSA berenkode base64 yang Anda hasilkan dengan menggunakan utilitas yang dijelaskan dalamthe section called “Prasyarat”, dan`CHEF_AUTOMATE_ADMIN_PASSWORD`, kata sandi antara delapan dan 32 karakter yang Anda buat. Untuk informasi selengkapnya tentang `CHEF_AUTOMATE_ADMIN_PASSWORD`, lihat Buat server Chef Automate dengan menggunakan AWS CLI. Anda dapat memberikan pointer ke file PEM yang berisi kunci penting Anda sebagai nilai `PivotalKey` parameter, seperti yang ditunjukkan pada contoh. Jika nilai untuk `CHEF_AUTOMATE_ADMIN_PASSWORD` dan tidak `CHEF_AUTOMATE_PIVOTAL_KEY` ditentukan dalam template Anda, Anda harus memberikan nilai dalam AWS CLI perintah Anda.

```
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"
```

Berikut ini adalah contoh yang mencakup nilai sampel untuk `CHEF_AUTOMATE_ADMIN_PASSWORD` dan `CHEF_AUTOMATE_PIVOTAL_KEY` atribut. Jalankan perintah serupa jika Anda tidak menentukan nilai untuk atribut ini di AWS CloudFormation template Anda.

```
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. Saat pembuatan tumpukan selesai, buka halaman Properties untuk server baru di AWS OpsWorks for Chef Automate konsol, dan unduh starter kit. Mengunduh starter kit baru akan mengatur ulang kata sandi administrator dasbor Chef Automate.
5. Jika server Anda akan menggunakan domain kustom, sertifikat, dan kunci pribadi, ikuti langkah-langkah untuk mengkonfigurasi `knife.rb`(Opsional) Konfigurasi `knife` untuk Bekerja dengan Domain Kustom, dan kemudian lanjutkan ke langkah 7.
6. Untuk menggunakan `knife` perintah di server baru, perbarui pengaturan file `knife.rb` konfigurasi Chef. `knife.rb`File contoh disertakan dengan starter kit. Contoh berikut menunjukkan cara mengatur `knife.rb` pada server yang tidak menggunakan domain kustom. Jika Anda menggunakan domain kustom, lihat (Opsional) Konfigurasi `knife` untuk Bekerja dengan Domain Kustom petunjuk `knife` konfigurasi.

   - Ganti *`ENDPOINT`* dengan nilai endpoint server. Ini adalah bagian dari output dari operasi pembuatan tumpukan. Anda bisa mendapatkan endpoint dengan menjalankan perintah berikut.

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

   - Ganti *`key_pair_file.pem`* dalam `client_key` konfigurasi dengan nama file PEM yang berisi `CHEF_AUTOMATE_PIVOTAL_KEY` yang Anda gunakan untuk membuat server Anda.

     ```
     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. Ketika proses pembuatan server selesai, lanjutkan kethe section called “Selesaikan konfigurasi dan unggah buku masak”. Jika pembuatan tumpukan gagal, tinjau pesan kesalahan yang ditampilkan di konsol untuk membantu Anda menyelesaikan masalah. Untuk informasi selengkapnya tentang pemecahan masalah kesalahan di AWS CloudFormation tumpukan, lihat Memecahkan Masalah Kesalahan di Panduan Pengguna.AWS CloudFormation

Memperbarui AWS OpsWorks for Chef Automate Server untuk Menggunakan Domain Kustom

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara memperbarui AWS OpsWorks for Chef Automate server yang ada untuk menggunakan domain dan sertifikat khusus dengan menggunakan cadangan server untuk membuat server baru. Pada dasarnya, Anda menyalin server AWS OpsWorks for Chef Automate 2.0 yang ada dengan membuat server baru dari cadangan, kemudian mengkonfigurasi server baru untuk menggunakan domain kustom, sertifikat, dan kunci pribadi.

Topik

- Prasyarat

- Batasan
- Memperbarui Server untuk Menggunakan Domain Kustom
- Lihat Juga

Prasyarat

Berikut ini adalah persyaratan untuk memperbarui AWS OpsWorks for Chef Automate server yang ada untuk menggunakan domain dan sertifikat khusus.

- Server yang ingin Anda perbarui (atau salin) harus menjalankan Chef Automate 2.0.
- Tentukan cadangan mana yang ingin Anda gunakan untuk membuat server baru. Anda harus memiliki setidaknya satu cadangan yang tersedia dari server yang ingin Anda perbarui. Untuk informasi selengkapnya tentang backup di AWS OpsWorks for Chef Automate, lihat. Cadangkan AWS OpsWorks for Chef Automate Server
- Siapkan peran layanan dan profil instance ARNs yang Anda gunakan untuk membuat server yang ada yang merupakan sumber cadangan Anda.
- Pastikan bahwa Anda menjalankan rilis terbaru dari AWS CLI. Untuk informasi selengkapnya tentang memperbarui AWS CLI alat Anda, lihat Menginstal AWS CLI di Panduan Pengguna Antarmuka Baris Perintah AWS.

Batasan

Ketika Anda memperbarui server yang ada dengan membuat server baru dari cadangan, server baru tidak dapat persis sama dengan AWS OpsWorks for Chef Automate server yang ada.

- Anda hanya dapat menyelesaikan prosedur ini dengan menggunakan AWS CLI atau salah satu dari AWS SDKs. Anda tidak dapat membuat server baru dari cadangan dengan menggunakan file AWS Management Console.
- Server baru tidak dapat menggunakan nama yang sama dengan server yang ada di dalam akun, dan dalam Wilayah AWS. Nama harus berbeda dari server yang ada yang Anda gunakan sebagai sumber cadangan.
- Node yang dilampirkan ke server yang ada tidak dikelola oleh server baru. Anda harus melakukan salah satu dari yang berikut ini.
  - Lampirkan node yang berbeda, karena node tidak dapat dikelola oleh lebih dari satu server Chef Automate.

- Migrasikan node dari server yang ada (sumber cadangan) ke server baru dan endpoint domain kustom baru. Untuk informasi selengkapnya tentang cara memigrasi node, lihat di dokumentasi Chef.

Memperbarui Server untuk Menggunakan Domain Kustom

Untuk memperbarui server Chef Automate 2.0 yang ada, Anda membuat salinannya dengan menjalankan `create-server` perintah, menambahkan parameter untuk menentukan cadangan, domain kustom, sertifikat khusus, dan kunci pribadi khusus.

1. Jika Anda tidak memiliki peran layanan atau profil instans yang ARNs tersedia untuk ditentukan dalam `create-server` perintah Anda, ikuti langkah 1-5 Buat server Chef Automate dengan menggunakan AWS CLI untuk membuat peran layanan dan profil instance yang dapat Anda gunakan.
2. Jika Anda belum melakukannya, temukan cadangan server Chef Automate 2.0 yang ada di mana Anda ingin mendasarkan server baru dengan domain khusus. Jalankan perintah berikut untuk menampilkan informasi tentang semua AWS OpsWorks for Chef Automate cadangan di akun Anda, dan di suatu wilayah. Pastikan untuk mencatat ID cadangan yang ingin Anda gunakan.

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

3. Buat AWS OpsWorks for Chef Automate server dengan menjalankan `create-server` perintah.

   - `--engine`Nilai adalah`ChefAutomate`, `--engine-model` adalah`Single`, dan `--engine-version` adalah`12`.
   - Nama server harus unik di dalam AWS akun Anda, di setiap wilayah. Nama server harus dimulai dengan huruf; kemudian huruf, angka, atau tanda hubung (-) diperbolehkan, hingga maksimum 40 karakter.
   - Gunakan ARN profil instance dan ARN peran layanan dari langkah 1.
   - Jenis contoh yang valid adalah`m5.large`,`r5.xlarge`, ataur`5.2xlarge`. Untuk informasi selengkapnya tentang spesifikasi jenis instans ini, lihat Jenis Instance di Panduan EC2 Pengguna Amazon.
   - `--engine-attributes`Parameternya opsional; jika Anda tidak menentukan satu atau kedua nilai, proses pembuatan server menghasilkan nilai untuk Anda. Jika Anda menambahkan`--`

`engine-attributes`, tentukan `CHEF_AUTOMATE_PIVOTAL_KEY` nilai yang Anda hasilkan di Langkah 2, a`CHEF_AUTOMATE_ADMIN_PASSWORD`, atau keduanya.

Jika Anda tidak menetapkan nilai untuk`CHEF_AUTOMATE_ADMIN_PASSWORD`, kata sandi dibuat dan dikembalikan sebagai bagian dari `create-server` respons. Anda juga dapat mengunduh starter kit lagi di konsol, yang meregenerasi kata sandi ini. Panjang kata sandi minimal delapan karakter, dan maksimum 32. Kata sandi dapat berisi huruf, angka, dan karakter khusus (`!/@#$%^+=_`). Sandi harus berisi setidaknya satu huruf kecil, satu huruf besar, satu angka, dan satu karakter khusus.

- Sebuah key pair SSH bersifat opsional, tetapi dapat membantu Anda terhubung ke server Chef Automate Anda jika Anda perlu mengatur ulang kata sandi administrator dasbor Chef Automate. Untuk informasi selengkapnya tentang membuat key pair SSH, lihat Amazon EC2 Key Pairs di Panduan EC2 Pengguna Amazon.
- Untuk menggunakan domain kustom, tambahkan parameter berikut ke perintah Anda. Jika tidak, proses pembuatan server Chef Automate secara otomatis menghasilkan titik akhir untuk Anda. Ketiga parameter tersebut diperlukan untuk mengonfigurasi domain khusus. Untuk informasi tentang persyaratan tambahan untuk menggunakan parameter ini, lihat CreateServerdi Referensi AWS OpsWorks CM API.
  - `--custom-domain`- Sebuah endpoint publik opsional dari server, sepertihttps://`aws.my-company.com`.
  - `--custom-certificate`- Sertifikat HTTPS berformat PEM. Nilai dapat menjadi sertifikat tunggal yang ditandatangani sendiri, atau rantai sertifikat.
  - `--custom-private-key`- Kunci pribadi dalam format PEM untuk menghubungkan ke server dengan menggunakan HTTPS. Kunci privat tidak boleh dienkripsi; tidak dapat dilindungi oleh kata sandi atau frasa sandi.
- Pemeliharaan sistem mingguan diperlukan. Nilai yang valid harus ditentukan dalam format berikut:`DDD:HH:MM`. Waktu yang ditentukan adalah dalam Waktu Universal Terkoordinasi (UTC). Jika Anda tidak menentukan nilai untuk`--preferred-maintenance-window`, nilai default adalah acak, periode satu jam pada hari Selasa, Rabu, atau Jumat.
- Nilai yang valid untuk `--preferred-backup-window` harus ditentukan dalam salah satu format berikut: `HH:MM` untuk backup harian, atau `DDD:HH:MM` untuk backup mingguan. Waktu yang ditentukan adalah dalam UTC. Nilai default adalah waktu mulai acak harian. Untuk memilih keluar dari backup otomatis, tambahkan parameter `--disable-automated-backup` sebagai gantinya.

- Untuk `--security-group-ids`, masukkan satu atau lebih grup keamanan IDs, dipisahkan oleh spasi.
- Untuk `--subnet-ids`, masukkan subnet ID.
- Untuk `--backup-id`, masukkan ID cadangan yang Anda salin di langkah 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
```

Contoh berikut membuat server Chef Automate yang menggunakan domain kustom.

```
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 membutuhkan waktu sekitar 15 menit untuk membuat server baru. Dalam output `create-server` perintah, salin nilai `Endpoint` atribut. Berikut adalah contohnya.

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

   Jangan mengabaikan output dari `create-server` perintah atau menutup sesi shell Anda, karena output dapat berisi informasi penting yang tidak ditampilkan lagi. Untuk mendapatkan kata sandi dan starter kit dari `create-server` hasil, lanjutkan ke langkah berikutnya.

5. Jika Anda memilih untuk AWS OpsWorks for Chef Automate membuat kunci dan kata sandi untuk Anda, Anda dapat mengekstraknya dalam format yang dapat digunakan dari `create-server` hasil dengan menggunakan prosesor JSON seperti jq. Setelah Anda menginstal jq, Anda dapat menjalankan perintah berikut untuk mengekstrak kunci penting, kata sandi administrator dasbor Chef Automate, dan starter kit. Jika Anda tidak memberikan kunci dan kata sandi penting Anda sendiri di Langkah 3, pastikan untuk menyimpan kunci penting dan kata sandi administrator yang diekstraksi di lokasi yang nyaman namun aman.

   ```
   #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. Secara opsional, jika Anda tidak mengekstrak starter kit dari hasil `create-server` perintah, Anda dapat mengunduh starter kit baru dari halaman Properties server di AWS OpsWorks for Chef Automate konsol. Mengunduh starter kit baru akan mengatur ulang kata sandi administrator dasbor Chef Automate.

7. Buat entri CNAME di alat manajemen DNS perusahaan Anda untuk mengarahkan domain kustom Anda ke AWS OpsWorks for Chef Automate titik akhir yang Anda salin pada langkah 4. Anda tidak dapat menjangkau atau masuk ke server sampai Anda menyelesaikan langkah ini.

8. Ketika proses pembuatan server selesai, lanjutkan kethe section called “Selesaikan konfigurasi dan unggah buku masak”.

Lihat Juga

- Buat server Chef Automate dengan menggunakan AWS CLI
- Memulihkan AWS OpsWorks for Chef Automate Server dari Cadangan
- CreateServerdi Referensi AWS OpsWorks CM API
- create-serverdalam Referensi AWS CLI Perintah

Regenerasi starter kit untuk server AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Starter kit untuk AWS OpsWorks for Chef Automate berisi file README dengan contoh, file `knife.rb` konfigurasi, dan kunci pribadi untuk pengguna primer, atau penting,. Sebuah key pair baru dihasilkan—dan kunci lama disetel ulang—setiap kali Anda mengunduh starter kit. Anda dapat meregenerasi starter kit untuk AWS OpsWorks for Chef Automate server dengan salah satu dari dua cara:

- Di AWS OpsWorks konsol, pada menu Tindakan halaman detail untuk AWS OpsWorks for Chef Automate server. Anda diminta untuk mengonfirmasi apakah Anda ingin membuat ulang dan mengatur ulang kunci penting lama.
- Dengan menjalankan perintah di AWS CLI.

Untuk informasi lebih lanjut tentang cara menggunakan starter kit, lihatKonfigurasikan Server Chef Menggunakan Starter Kit.

Regenerasi AWS OpsWorks for Chef Automate starter kit dengan AWS CLI

> **Note**
>
> Saat Anda membuat ulang starter kit, Anda juga membuat ulang dan mengatur ulang key pair otentikasi untuk server Chef Automate Anda, dan menghapus key pair saat ini.

Regenerasi starter kit dengan menjalankan update-server-engine-attributesperintah. Dalam AWS CLI sesi, jalankan perintah berikut. Tentukan nama server Anda sebagai nilai`--server-name`. Untuk menetapkan kunci publik Anda sendiri sebagai nilai`CHEF_AUTOMATE_PIVOTAL_KEY`, tentukan nilai kunci publik di`--attribute-value`. Jika tidak, atur `--attribute-value` ke null.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name server_name \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value your_public_key
```

Perintah berikut adalah contoh yang menentukan nilai kunci publik yang administrator server ingin menggunakan.

```
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-----"
```

Perintah berikut adalah contoh yang memungkinkan AWS OpsWorks for Chef Automate regenerasi kunci publik.

```
aws opsworks-cm update-server-engine-attributes \
   --server-name your-test-server \
   --attribute-name "CHEF_AUTOMATE_PIVOTAL_KEY" \
   --attribute-value null
```

Output dari perintah ini adalah informasi tentang server, dan file ZIP yang dikodekan base64. File ZIP berisi starter kit Chef, yang mencakup README, file konfigurasi, dan kunci pribadi RSA yang diperlukan. Simpan file ini, unzip, dan kemudian ubah ke direktori tempat Anda membuka ritsleting konten file. Dari direktori ini, Anda dapat menjalankan `knife` perintah.

Bekerja dengan Tag pada AWS OpsWorks for Chef Automate Sumber Daya

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Tag adalah kata atau frasa yang bertindak sebagai metadata untuk mengidentifikasi dan mengatur sumber daya AWS Anda. Di AWS OpsWorks for Chef Automate, sumber daya dapat memiliki hingga 50 tag yang diterapkan pengguna. Setiap tag terdiri dari kunci dan satu nilai opsional. Anda dapat menerapkan tag ke sumber daya berikut di AWS OpsWorks for Chef Automate:

- AWS OpsWorks for Chef Automate server
- Cadangan server AWS OpsWorks for Chef Automate

Tag pada AWS sumber daya dapat membantu Anda melacak biaya, mengontrol akses ke sumber daya, mengelompokkan sumber daya untuk mengotomatisasi tugas, atau mengatur sumber daya berdasarkan tujuan atau tahap siklus hidup. Untuk informasi selengkapnya tentang manfaat tag, lihat Strategi Penandaan AWS di AWS Answers dan Menggunakan Tag Alokasi Biaya di AWS Manajemen Penagihan dan Biaya Panduan Pengguna.

Untuk menggunakan tag untuk mengontrol akses ke AWS OpsWorks for Chef Automate server atau cadangan, Anda membuat atau mengedit pernyataan kebijakan di AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan AWS Identity and Access Management Pengguna.

Saat Anda menerapkan tag ke AWS OpsWorks for Chef Automate server, tag juga diterapkan ke cadangan server, bucket Amazon S3 yang menyimpan cadangan, instance EC2 Amazon server, rahasia untuk server yang disimpan AWS Secrets Manager, dan alamat IP Elastis yang digunakan oleh server. Tag tidak disebarkan ke AWS CloudFormation tumpukan yang AWS OpsWorks digunakan untuk membuat server Anda.

Topik

- Bagaimana Tag Bekerja di AWS OpsWorks for Chef Automate
- Tambahkan dan Kelola Tag di AWS OpsWorks for Chef Automate (Konsol)
- Tambahkan dan Kelola Tag di AWS OpsWorks for Chef Automate (CLI)
- Lihat Juga

Bagaimana Tag Bekerja di AWS OpsWorks for Chef Automate

Dalam rilis ini, Anda dapat menambahkan dan mengelola tag dengan menggunakan AWS OpsWorks CM API atau file AWS Management Console. AWS OpsWorks CM juga mencoba menambahkan tag yang Anda tambahkan ke server ke AWS sumber daya yang terkait dengan server, termasuk EC2 instance, rahasia di Secrets Manager, alamat IP Elastis, grup keamanan, bucket S3, dan backup. Tabel berikut memberikan ikhtisar tentang bagaimana Anda menambahkan dan mengelola tag di AWS OpsWorks for Chef Automate.

Tindakan	Apa yang harus digunakan
Tambahkan tag ke AWS OpsWorks for Chef Automate server baru atau cadangan yang Anda buat secara manual.	• Pilih Create Chef Automate server dan tambahkan tag pada halaman Configure advanced settings. • Pilih Buat cadangan di halaman Cadangan untuk server yang ada, dan tambahkan tag pada halaman Buat cadangan Chef Automate 2 server Anda. • Tambahkan `Tags` parameter ke `CreateBackup` perintah `CreateServer` atau.
Lihat tag pada sumber daya.	• Pada halaman detail untuk server Anda, pilih Tag di panel navigasi. • Pada halaman Backup untuk server Anda, pilih cadangan, lalu pilih Edit cadangan. • Jalankan perintah `ListTagsForResource`.

Tindakan	Apa yang harus digunakan
Tambahkan tag ke AWS OpsWorks for Chef Automate server yang ada atau cadangan, terlepas dari apakah cadangan dibuat secara manual atau otomatis.	• Pada halaman detail untuk server Anda, pilih Tag di panel navigasi, lalu pilih Edit. • Pada halaman Backup untuk server Anda, pilih cadangan, lalu pilih Edit cadangan. • Jalankan perintah `TagResource` .
Hapus tag dari sumber daya.	• Pada halaman detail untuk server Anda, pilih Tag di panel navigasi, lalu pilih Edit. Pilih X di samping tag yang ingin Anda hapus. • Pada halaman Backup untuk server Anda, pilih cadangan, lalu pilih Edit cadangan. Pilih X di samping tag yang ingin Anda hapus. • Jalankan perintah `UntagResource` .

`DescribeServers`dan `DescribeBackups` tanggapan tidak termasuk informasi tag. Untuk menampilkan tag, gunakan `ListTagsForResource` API.

Tambahkan dan Kelola Tag di AWS OpsWorks for Chef Automate (Konsol)

Prosedur di bagian ini dilakukan di AWS Management Console.

Jika Anda menambahkan tag, kunci tag tidak bisa kosong. Kuncinya bisa maksimal 127 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: Nilai tag `+ - = . _ : / @` adalah opsional. Anda dapat menambahkan tag yang memiliki kunci, tetapi tidak ada nilai. Nilai dapat maksimal 255 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: `+ - = . _ : / @`

Topik

- Tambahkan Tag ke AWS OpsWorks for Chef Automate Server Baru (Konsol)
- Tambahkan Tag ke Cadangan Baru (Konsol)
- Menambahkan atau Melihat Tag di Server yang Ada (Konsol)
- Menambahkan atau Melihat Tag pada Cadangan yang Ada (Konsol)
- Hapus Tag dari Server (Konsol)

- Hapus Tag dari Backup (Konsol)

Tambahkan Tag ke AWS OpsWorks for Chef Automate Server Baru (Konsol)

1. Pastikan untuk menyelesaikan prasyarat apa pun untuk membuat server. AWS OpsWorks for Chef Automate
2. Ikuti langkah 1-10 inBuat Server Otomatis Koki.
3. Setelah Anda menentukan pengaturan cadangan otomatis, tambahkan tag di area Tag pada halaman Konfigurasikan pengaturan lanjutan. Anda dapat menambahkan maksimal 50 tag. Setelah selesai menambahkan tag, pilih Berikutnya.
4. Lanjutkan ke langkah 13Buat Server Otomatis Koki, dan tinjau pengaturan yang telah Anda pilih untuk server baru.

Tambahkan Tag ke Cadangan Baru (Konsol)

1. Di AWS OpsWorks for Chef Automate halaman beranda, pilih server Chef Automate yang ada.
2. Dari halaman detail server, pilih Backup di panel navigasi.
3. Pada halaman Backup, pilih Buat cadangan.
4. Tambahkan tag. Pilih Buat ketika Anda selesai menambahkan tag.

Menambahkan atau Melihat Tag di Server yang Ada (Konsol)

1. Di AWS OpsWorks for Chef Automate halaman beranda, pilih server Chef Automate yang ada untuk membuka halaman detailnya.
2. Pilih Tag di panel navigasi, atau di bagian bawah halaman detail, pilih Lihat semua tag.
3. Pada halaman Tag, pilih Edit.
4. Tambahkan atau edit tag di server. Setelah selesai, pilih Simpan.

> **Note**
>
> Ketahuilah bahwa mengubah tag di server Chef Automate Anda juga mengubah tag pada sumber daya yang terkait dengan server, seperti EC2 instance, alamat IP Elastis, grup keamanan, bucket S3, dan cadangan.

Menambahkan atau Melihat Tag pada Cadangan yang Ada (Konsol)

1. Di AWS OpsWorks for Chef Automate halaman beranda, pilih server Chef Automate yang ada untuk membuka halaman detailnya.
2. Pilih Cadangan di panel navigasi, atau di area Pencadangan terbaru pada halaman detail, pilih Lihat semua cadangan.
3. Pada halaman Cadangan, pilih cadangan untuk dikelola, lalu pilih Edit cadangan.
4. Tambahkan atau edit tag pada cadangan. Pilih Perbarui saat Anda selesai.

Hapus Tag dari Server (Konsol)

1. Di AWS OpsWorks for Chef Automate halaman beranda, pilih server Chef Automate yang ada untuk membuka halaman detailnya.
2. Pilih Tag di panel navigasi, atau di bagian bawah halaman detail, pilih Lihat semua tag.
3. Pada halaman Tag, pilih Edit.
4. Pilih X di samping tag untuk menghapus tag. Setelah selesai, pilih Simpan.

> **Note**
>
> Ketahuilah bahwa mengubah tag di server Chef Automate Anda juga mengubah tag pada sumber daya yang terkait dengan server, seperti EC2 instance, alamat IP Elastis, grup keamanan, bucket S3, dan cadangan.

Hapus Tag dari Backup (Konsol)

1. Di AWS OpsWorks for Chef Automate halaman beranda, pilih server Chef Automate yang ada untuk membuka halaman detailnya.
2. Pilih Cadangan di panel navigasi, atau di area Pencadangan terbaru pada halaman detail, pilih Lihat semua cadangan.
3. Pada halaman Cadangan, pilih cadangan untuk dikelola, lalu pilih Edit cadangan.
4. Pilih X di samping tag untuk menghapus tag. Pilih Perbarui saat Anda selesai.

Tambahkan dan Kelola Tag di AWS OpsWorks for Chef Automate (CLI)

Prosedur di bagian ini dilakukan di AWS CLI. Pastikan bahwa Anda menjalankan rilis terbaru AWS CLI sebelum Anda mulai bekerja dengan tag. Untuk informasi selengkapnya tentang menginstal atau memperbarui AWS CLI, lihat Menginstal AWS CLI di Panduan AWS Command Line Interface Pengguna.

Jika Anda menambahkan tag, kunci tag tidak bisa kosong. Kuncinya bisa maksimal 127 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: Nilai tag `+ - = . _ : / @` adalah opsional. Anda dapat menambahkan tag yang memiliki kunci, tetapi tidak ada nilai. Nilai dapat maksimal 255 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: `+ - = . _ : / @`

Topik

- Tambahkan Tag ke AWS OpsWorks for Chef Automate Server Baru (CLI)
- Tambahkan Tag ke Cadangan Baru (CLI)
- Tambahkan Tag ke Server atau Cadangan yang Ada (CLI)
- Daftar Sumber Daya Tag
- Hapus Tag dari Sumber Daya

Tambahkan Tag ke AWS OpsWorks for Chef Automate Server Baru (CLI)

Anda dapat menggunakan AWS CLI untuk menambahkan tag saat Anda membuat AWS OpsWorks for Chef Automate server. Prosedur ini tidak menjelaskan secara lengkap cara membuat server. Untuk informasi rinci tentang cara membuat AWS OpsWorks for Chef Automate server dengan menggunakan AWS CLI lihat Buat server Chef Automate dengan menggunakan AWS CLI di panduan ini. Anda dapat menambahkan hingga 50 tag ke server.

1. Pastikan untuk menyelesaikan prasyarat apa pun untuk membuat server. AWS OpsWorks for Chef Automate
2. Selesaikan langkah 1-5 dari. Buat server Chef Automate dengan menggunakan AWS CLI
3. Untuk langkah 6, saat Anda menjalankan `create-server` perintah, tambahkan `--tags` parameter ke perintah, seperti yang ditunjukkan pada contoh berikut.

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

Berikut ini adalah contoh yang hanya menampilkan bagian tag dari `create-server` perintah.

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

4. Selesaikan langkah-langkah selebihnya di Buat server Chef Automate dengan menggunakan AWS CLI. Untuk memverifikasi bahwa tag Anda telah ditambahkan ke server baru, ikuti langkah-langkah Daftar Sumber Daya Tag dalam topik ini.

Tambahkan Tag ke Cadangan Baru (CLI)

Anda dapat menggunakan AWS CLI untuk menambahkan tag saat Anda membuat cadangan manual AWS OpsWorks for Chef Automate server yang baru. Prosedur ini tidak menjelaskan secara lengkap cara membuat cadangan manual. Untuk informasi terperinci tentang cara membuat cadangan manual, lihat “Untuk melakukan pencadangan manual di AWS CLI" diCadangkan AWS OpsWorks for Chef Automate Server. Anda dapat menambahkan hingga 50 tag ke cadangan. Jika server memiliki tag, backup baru secara otomatis ditandai dengan tag server.

Secara default, saat Anda membuat AWS OpsWorks for Chef Automate server baru, pencadangan otomatis diaktifkan. Anda dapat menambahkan tag ke cadangan otomatis dengan menjalankan `tag-resource` perintah, yang dijelaskan Tambahkan Tag ke Server atau Cadangan yang Ada (CLI) dalam topik ini.

- Untuk menambahkan tag ke cadangan manual saat Anda membuat cadangan, jalankan perintah berikut. Hanya bagian tag dari perintah yang ditampilkan. Untuk contoh `create-backup` perintah lengkap, lihat “Untuk melakukan pencadangan manual di AWS CLI" inCadangkan AWS OpsWorks for Chef Automate Server.

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

  Contoh berikut hanya menunjukkan bagian tag dari `create-backup` perintah.

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

Tambahkan Tag ke Server atau Cadangan yang Ada (CLI)

Anda dapat menjalankan `tag-resource` perintah untuk menambahkan tag ke AWS OpsWorks for Chef Automate server atau cadangan yang ada (apakah cadangan dibuat secara otomatis atau manual). Tentukan Nomor Sumber Daya Amazon (ARN) dari sumber daya target untuk menambahkan tag ke dalamnya.

1. Untuk mendapatkan ARN dari sumber daya yang ingin Anda terapkan tag:

   - Untuk server, jalankan`describe-servers --server-name` *`server_name`*. Hasil perintah menunjukkan server ARN.
   - Untuk cadangan, jalankan`describe-backups --backup-id` *`backup_ID`*. Hasil perintah menunjukkan ARN cadangan. Anda juga dapat menjalankan `describe-backups --server-name` *`server_name`* untuk menampilkan informasi tentang semua cadangan untuk server tertentu AWS OpsWorks for Chef Automate .

   Contoh berikut hanya menunjukkan hasil `describe-servers --server-name opsworks-cm-test` perintah. `ServerArn` `ServerArn`Nilai ditambahkan ke `tag-resource` perintah untuk menambahkan tag ke server.

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

2. Jalankan `tag-resource` perintah dengan ARN yang Anda kembalikan pada langkah 1.

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

   Berikut adalah contohnya.

```
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. Untuk memverifikasi bahwa tag berhasil ditambahkan, lanjutkan ke prosedur berikutnya,Daftar Sumber Daya Tag.

Daftar Sumber Daya Tag

Anda dapat menjalankan `list-tags-for-resource` perintah untuk menampilkan tag yang dilampirkan ke AWS OpsWorks for Chef Automate server atau cadangan. Tentukan ARN dari sumber daya target untuk melihat tag-nya.

1. Untuk mendapatkan ARN dari sumber daya yang ingin Anda cantumkan tag:
   - Untuk server, jalankan`describe-servers --server-name` *`server_name`*. Hasil perintah menunjukkan server ARN.
   - Untuk cadangan, jalankan`describe-backups --backup-id` *`backup_ID`*. Hasil perintah menunjukkan ARN cadangan. Anda juga dapat menjalankan `describe-backups --server-name` *`server_name`* untuk menampilkan informasi tentang semua cadangan untuk server tertentu AWS OpsWorks for Chef Automate .
2. Jalankan `list-tags-for-resource` perintah dengan ARN yang Anda kembalikan pada langkah 1.

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

   Berikut adalah contohnya.

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

   Jika ada tag pada sumber daya, perintah mengembalikan hasil seperti berikut ini.

   ```
   {
       "Tags": [
           {
               "Key": "Stage",
               "Value": "Production"
   ```

```
        },
        {
            "Key": "Department",
            "Value": "Marketing"
        }
    ]
}
```

Hapus Tag dari Sumber Daya

Anda dapat menjalankan `untag-resource` perintah untuk menghapus tag dari AWS OpsWorks for Chef Automate server atau cadangan. Jika sumber daya dihapus, tag pada sumber daya juga dihapus. Tentukan Nomor Sumber Daya Amazon (ARN) dari sumber daya target untuk menghapus tag darinya.

1. Untuk mendapatkan ARN dari sumber daya dari mana Anda ingin menghapus tag:

   - Untuk server, jalankan`describe-servers --server-name` *`server_name`*. Hasil perintah menunjukkan server ARN.
   - Untuk cadangan, jalankan`describe-backups --backup-id` *`backup_ID`*. Hasil perintah menunjukkan ARN cadangan. Anda juga dapat menjalankan `describe-backups --server-name` *`server_name`* untuk menampilkan informasi tentang semua cadangan untuk server tertentu AWS OpsWorks for Chef Automate .

2. Jalankan `untag-resource` perintah dengan ARN yang Anda kembalikan pada langkah 1. Tentukan hanya tag yang ingin Anda hapus.

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

   Dalam contoh ini, `untag-resource` perintah hanya menghapus tag dengan kunci dari `Stage` dan nilai`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. Untuk memverifikasi bahwa tag berhasil dihapus, ikuti langkah-langkah Daftar Sumber Daya Tag dalam topik ini.

Lihat Juga

- Buat server Chef Automate dengan menggunakan AWS CLI
- Cadangkan AWS OpsWorks for Chef Automate Server
- Strategi Penandaan AWS
- Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan AWS Identity and Access Management Pengguna
- Menggunakan Tanda Alokasi Biaya dalam AWS Manajemen Penagihan dan Biaya Panduan Pengguna
- CreateBackupdi Referensi AWS OpsWorks CM API
- CreateServerdi Referensi AWS OpsWorks CM API
- TagResourcedi Referensi AWS OpsWorks CM API
- ListTagsForResourcedi Referensi AWS OpsWorks CM API
- UntagResourcedi Referensi AWS OpsWorks CM API

Membuat cadangan dan Memulihkan AWS OpsWorks for Chef Automate Server

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara mencadangkan dan memulihkan AWS OpsWorks for Chef Automate server, dan cara menghapus cadangan.

Topik

- Cadangkan AWS OpsWorks for Chef Automate Server
- Memulihkan AWS OpsWorks for Chef Automate Server dari Cadangan

Cadangkan AWS OpsWorks for Chef Automate Server

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Anda dapat menentukan backup AWS OpsWorks for Chef Automate server harian atau mingguan, dan meminta layanan menyimpan backup di Amazon Simple Storage Service (Amazon S3) atas nama Anda. Atau, Anda dapat membuat cadangan manual sesuai permintaan.

Karena backup disimpan di Amazon S3, mereka dikenakan biaya tambahan. Anda dapat menentukan periode retensi backup hingga 30 generasi. Anda dapat mengirimkan permintaan layanan agar batas tersebut diubah dengan menggunakan saluran AWS dukungan. Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Cara Menghapus S3 Bucket? .

Anda dapat menambahkan tag ke backup server. AWS OpsWorks for Chef Automate Jika Anda telah menambahkan tag ke AWS OpsWorks for Chef Automate server, backup otomatis server mewarisi tag tersebut. Untuk informasi selengkapnya tentang cara menambahkan dan mengelola tag pada backup, lihat Bekerja dengan Tag pada AWS OpsWorks for Chef Automate Sumber Daya di panduan ini.

Topik

- Pencadangan Otomatis
- Backup Manual
- Hapus backup

Pencadangan Otomatis

Ketika Anda mengkonfigurasi AWS OpsWorks for Chef Automate server Anda, Anda memilih backup otomatis atau manual. AWS OpsWorks for Chef Automate memulai pencadangan otomatis selama

satu jam dan pada hari yang Anda pilih di bagian Pencadangan otomatis pada halaman Konfigurasi pengaturan lanjutan dari Pengaturan. Setelah server Anda online, Anda dapat mengubah pengaturan cadangan dengan melakukan langkah-langkah berikut, baik dari ubin server di halaman beranda server Chef Automate, atau di halaman Properti server.

Untuk mengubah setelan pencadangan otomatis

1. Di menu Tindakan ubin server di halaman beranda server Chef, pilih Ubah pengaturan
2. Untuk mematikan pencadangan otomatis, pilih Tidak untuk opsi Aktifkan pencadangan otomatis. Simpan perubahan Anda; Anda tidak perlu melanjutkan ke langkah berikutnya.
3. Di bagian Backup Otomatis, ubah frekuensi, waktu mulai, atau generasi yang akan disimpan. Simpan perubahan Anda.

Backup Manual

Anda dapat memulai pencadangan manual kapan saja di AWS Management Console, atau dengan menjalankan perintah AWS CLI create-backup. Pencadangan manual tidak termasuk dalam maksimum 30 generasi pencadangan otomatis yang disimpan; maksimal 10 cadangan manual disimpan, dan harus dihapus secara manual dari Amazon S3.

Anda dapat menambahkan tag ketika Anda membuat backup manual baru dari AWS OpsWorks for Chef Automate server. Untuk informasi selengkapnya tentang cara menambahkan tag saat membuat backup manual, lihatTambahkan Tag ke Cadangan Baru (CLI).

Untuk melakukan backup manual di AWS Management Console

1. Pada halaman server Chef Automate, pilih server yang ingin Anda backup.
2. Pada halaman properti server, di sebelah kiri panel navigasi, pilih backup.
3. Pilih Buat cadangan.
4. Pencadangan manual selesai ketika halaman menunjukkan tanda centang hijau di kolom Status cadangan.

Untuk melakukan backup manual di AWS CLI

- Untuk memulai backup manual, jalankan AWS CLI perintah berikut.

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

Hapus backup

Menghapus cadangan secara permanen menghapusnya dari bucket S3 tempat cadangan disimpan.

Untuk menghapus backup di AWS Management Console

1. Pada halaman server Chef Automate, pilih server yang ingin Anda backup.
2. Pada halaman properti server, di sebelah kiri panel navigasi, pilih backup.
3. Pilih backup yang ingin Anda hapus, dan pilih Hapus backup. Anda hanya dapat memilih satu backup dalam satu waktu.
4. Ketika Anda diminta untuk mengonfirmasi penghapusan, isi kotak centang untuk Hapus cadangan, yang disimpan dalam ember S3, lalu pilih Ya, Hapus.

Untuk menghapus backup di AWS CLI

- Untuk menghapus backup tersebut, jalankan AWS CLI perintah berikut, ganti `--backup-id` dengan ID backup yang ingin Anda hapus. Backup IDs dalam format*ServerName-yyyyMMddHHmmssSSS*. Misalnya, **`test-chef-server-20171218132604388`**.

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

Memulihkan AWS OpsWorks for Chef Automate Server dari Cadangan

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Setelah menelusuri cadangan yang tersedia, Anda dapat memilih titik waktu untuk memulihkan server Anda AWS OpsWorks for Chef Automate . Pencadangan server hanya berisi data persisten perangkat lunak manajemen konfigurasi (buku masak, node terdaftar, dll.). Melakukan restorasi server di tempat (yaitu memulihkan server yang ada AWS OpsWorks for Chef Automate ke EC2 instance baru) mendaftarkan ulang node yang terdaftar pada saat pencadangan yang Anda gunakan untuk memulihkan server, dan mengalihkan lalu lintas ke instance baru jika restorasi berhasil, dan status server yang dipulihkan adalah. AWS OpsWorks for Chef Automate `Healthy` Memulihkan ke AWS OpsWorks for Chef Automate server yang baru dibuat tidak mempertahankan koneksi node. Memulihkan server tidak memperbarui versi minor perangkat lunak Chef; itu menerapkan versi Chef yang sama dan data manajemen konfigurasi yang tersedia dalam cadangan yang Anda pilih.

Memulihkan server biasanya membutuhkan lebih banyak waktu daripada membuat server baru; waktu tergantung pada ukuran cadangan yang Anda pilih. Setelah restorasi selesai, EC2 contoh lama tetap dalam `Stopped` keadaan `Running` atau, tetapi hanya sementara. Itu akhirnya dihentikan.

Dalam rilis ini, Anda dapat menggunakan AWS CLI untuk memulihkan server Chef di AWS OpsWorks for Chef Automate.

> **Note**
>
> Anda juga dapat menjalankan perintah restore-server untuk mengubah jenis instans saat ini, atau untuk memulihkan atau mengatur kunci SSH Anda jika hilang atau disusupi.

Untuk memulihkan server dari backup

1. Dalam AWS CLI, jalankan perintah berikut untuk mengembalikan daftar backup yang tersedia dan mereka. IDs Buatlah sebuah catatan backup yang ingin Anda gunakan. Backup IDs dalam format*myServerName-yyyyMMddHHmmssSSS*.

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

2. Jalankan perintah berikut.

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

   Berikut adalah contohnya.

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

3. Tunggu sampai restorasi selesai.

Pemeliharaan Sistem di AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Pemeliharaan sistem wajib memastikan bahwa versi minor terbaru dari Chef Server dan Chef Automate Server, termasuk pembaruan keamanan, selalu berjalan di AWS OpsWorks for Chef Automate server. Pemeliharaan sistem diperlukan minimal seminggu sekali. Dengan menggunakan AWS CLI, Anda dapat mengkonfigurasi pemeliharaan otomatis harian, jika diinginkan. Anda juga dapat menggunakan AWS CLI untuk melakukan pemeliharaan sistem sesuai permintaan, selain pemeliharaan sistem terjadwal.

Ketika versi minor baru dari perangkat lunak Chef tersedia, pemeliharaan sistem dirancang untuk memperbarui versi minor Chef Automate dan Chef Server di server secara otomatis, segera setelah melewati pengujian AWS. AWS melakukan pengujian ekstensif untuk memverifikasi bahwa peningkatan Chef siap produksi dan tidak mengganggu lingkungan pelanggan yang ada, sehingga mungkin ada kelambatan antara rilis perangkat lunak Chef dan ketersediaannya untuk aplikasi yang ada untuk server Chef Automate. OpsWorks Untuk memperbarui versi minor yang tersedia dari perangkat lunak Chef sesuai permintaan, lihat Memulai pemeliharaan sistem sesuai permintaan di topik ini.

Pemeliharaan sistem meluncurkan instans baru dari cadangan yang dilakukan sebagai bagian dari proses pemeliharaan, yang membantu mengurangi risiko dari EC2 instans Amazon yang terdegradasi atau terganggu yang menjalani pemeliharaan berkala.

> **Important**
>
> Pemeliharaan sistem menghapus file atau konfigurasi khusus yang telah Anda tambahkan ke AWS OpsWorks for Chef Automate server. Untuk informasi selengkapnya tentang cara memperbaiki konfigurasi atau kehilangan file, lihat Memulihkan konfigurasi dan file khusus setelah pemeliharaan di topik ini.

Topik

- Memastikan node mempercayai Otoritas AWS OpsWorks Sertifikasi
- Mengkonfigurasi pemeliharaan sistem
- Memulai pemeliharaan sistem sesuai permintaan
- Memulihkan konfigurasi dan file khusus setelah pemeliharaan

Memastikan node mempercayai Otoritas AWS OpsWorks Sertifikasi

> **Note**
>
> Langkah-langkah di bagian ini tidak diperlukan jika Anda menggunakan domain dan sertifikat khusus dengan AWS OpsWorks for Chef Automate server Anda.

Node yang Anda kelola dengan AWS OpsWorks for Chef Automate server harus mengautentikasi dengan server dengan menggunakan sertifikat. Selama pemeliharaan sistem, AWS OpsWorks mengganti instance server, dan meregenerasi sertifikat baru melalui otoritas AWS OpsWorks sertifikat (CA). Untuk memulihkan komunikasi secara otomatis dengan node terkelola Anda setelah pemeliharaan selesai, node harus mempercayai AWS OpsWorks CA yang dikirimkan dengan starter kit, dan di-host di wilayah yang didukung oleh AWS OpsWorks for Chef Automate. Saat Anda menggunakan AWS OpsWorks CA untuk membangun kepercayaan antara node dan server, node terhubung kembali ke instance server baru setelah pemeliharaan. Jika Anda menambahkan EC2 node dengan menggunakan EC2 `userdata` skrip yang dijelaskan diTambahkan node secara otomatis di AWS OpsWorks for Chef Automate, node sudah dikonfigurasi untuk mempercayai AWS OpsWorks CA.

- Untuk node berbasis Linux, lokasi bucket S3 dari CA adalah. `https://opsworks-cm-${REGION}-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-`

`ca-2020-root.pem` CA AWS OpsWorks tepercaya harus disimpan di jalur`/etc/chef/opsworks-cm-ca-2020-root.pem`.

- Untuk node berbasis Windows, lokasi bucket S3 dari CA adalah. `https://opsworks-cm-$env:AWS_REGION-prod-default-assets.s3.amazonaws.com/misc/opsworks-cm-ca-2020-root.pem` AWS OpsWorks CA harus disimpan di folder root Chef; misalnya, `C:\chef\opsworks-cm-ca-2020-root.pem`

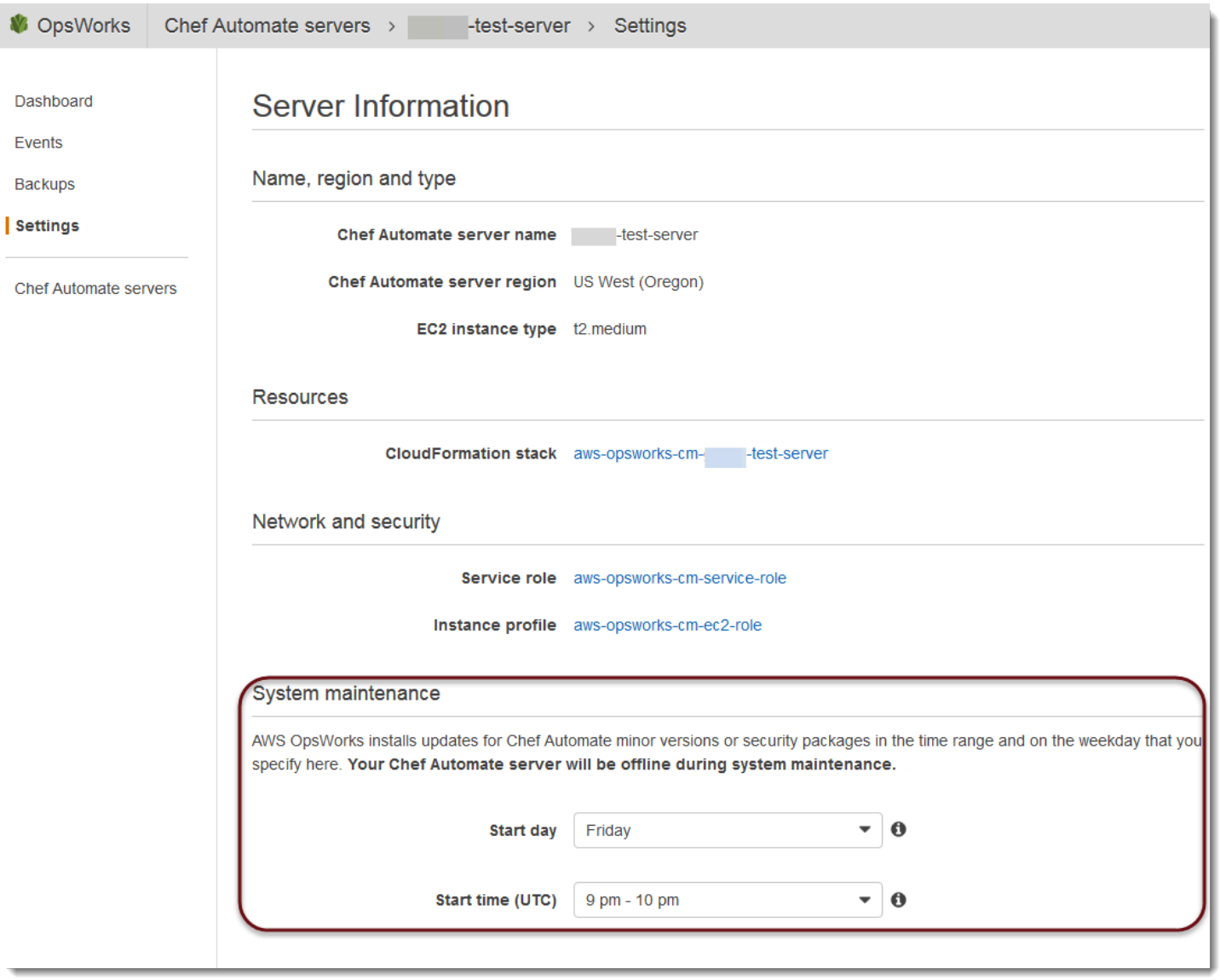
Di kedua jalur, variabel wilayah menyelesaikan salah satu dari yang berikut ini.

- `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`

Mengkonfigurasi pemeliharaan sistem

Saat Anda membuat AWS OpsWorks for Chef Automate server baru, Anda dapat mengonfigurasi hari kerja dan waktu, di Coordinated Universal Time (UTC), agar pemeliharaan sistem dimulai. Pemeliharaan dimulai selama jam yang Anda tentukan. Karena Anda harus mengharapkan server offline selama pemeliharaan sistem, pilih waktu permintaan server yang rendah dalam jam kantor reguler. Status server `UNDER_MAINTENANCE` saat pemeliharaan sedang berlangsung.

Anda juga dapat mengubah pengaturan pemeliharaan sistem pada AWS OpsWorks for Chef Automate server yang ada, dengan mengubah pengaturan di area pemeliharaan sistem halaman Pengaturan untuk server Anda, seperti yang ditunjukkan pada gambar berikut.



Di bagian Pemeliharaan sistem, atur hari dan jam yang Anda inginkan untuk memulai pemeliharaan sistem.

Mengkonfigurasi pemeliharaan sistem dengan menggunakan AWS CLI

Anda juga dapat mengonfigurasi waktu mulai otomatis pemeliharaan sistem dengan menggunakan AWS CLI. Ini AWS CLI memungkinkan Anda mengonfigurasi pemeliharaan otomatis harian, jika diinginkan, dengan menghilangkan awalan tiga karakter hari kerja.

Dalam sebuah `create-server` perintah, tambahkan `--preferred-maintenance-window` parameter ke perintah Anda, setelah menentukan persyaratan untuk membuat instance server (seperti jenis instance, ARN profil instance, dan ARN peran layanan). Dalam `create-server`

contoh berikut, `--preferred-maintenance-window` diatur ke`Mon:08:00`, artinya Anda telah mengatur pemeliharaan untuk dimulai setiap Senin pagi pukul 8:00 pagi. 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"
```

Dalam sebuah `update-server` perintah, Anda dapat memperbarui `--preferred-maintenance-window` nilainya sendiri, jika diinginkan. Dalam contoh berikut, jendela pemeliharaan diatur ke Jumat malam pukul 6:15 malam. UTC.

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

Untuk mengubah waktu mulai jendela pemeliharaan menjadi 18:15 UTC setiap hari, hilangkan awalan tiga karakter hari kerja, seperti yang ditunjukkan pada contoh berikut.

```
aws opsworks-cm update-server --server-name "shiny-kitchen" --preferred-maintenance-
window "18:15"
```

Untuk informasi selengkapnya tentang pengaturan jendela pemeliharaan sistem pilihan dengan menggunakan AWS CLI, lihat create-server dan update-server.

Memulai pemeliharaan sistem sesuai permintaan

Untuk memulai pemeliharaan sistem sesuai permintaan, di luar pemeliharaan otomatis mingguan atau harian yang dikonfigurasi, jalankan AWS CLI perintah berikut. Anda tidak dapat memulai pemeliharaan sesuai permintaan di. AWS Management Console

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

Untuk informasi selengkapnya tentang perintah ini, lihat pemeliharaan awal.

Memulihkan konfigurasi dan file khusus setelah pemeliharaan

Pemeliharaan sistem dapat menghapus atau mengubah file atau konfigurasi khusus yang telah Anda tambahkan ke AWS OpsWorks for Chef Automate server Anda.

Jika, setelah pemeliharaan dijalankan, server Chef Anda kehilangan file atau pengaturan yang Anda tambahkan dengan menggunakan `RunCommand` atau SSH, Anda dapat menggunakan Amazon Machine Image (AMI) untuk meluncurkan EC2 instans Amazon baru. AMIs tersedia yang dibangun dari konfigurasi pra-pemeliharaan server.

Instance baru berada dalam keadaan yang sama dengan server Chef sebelum pemeliharaan, dan harus menyertakan file dan pengaturan Anda yang hilang.

> **Important**
>
> Anda tidak dapat menggunakan instance baru untuk memulihkan server Anda; instance tidak dapat dijalankan sebagai server Chef. Anda dapat menggunakan instance hanya untuk memulihkan file dan pengaturan konfigurasi Anda.

Untuk meluncurkan EC2 instance dari AMI, di EC2 konsol Amazon, buka panduan Peluncuran, pilih My AMIs, lalu pilih AMI yang memiliki nama server Anda. Ikuti langkah-langkah EC2 wizard Amazon seperti yang Anda lakukan untuk peluncuran instans lainnya.

Pemindaian Kepatuhan di AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Pemindaian kepatuhan memungkinkan Anda melacak kepatuhan node terkelola di infrastruktur Anda berdasarkan kebijakan yang telah ditentukan sebelumnya, juga disebut aturan. Tampilan kepatuhan memungkinkan Anda secara teratur mengaudit aplikasi Anda untuk kerentanan dan konfigurasi yang tidak sesuai. Chef menawarkan lebih dari 100 profil kepatuhan yang telah ditentukan sebelumnya — kumpulan aturan yang berlaku untuk konfigurasi node tertentu—yang dapat Anda gunakan dalam pemindaian kepatuhan Anda. Anda juga dapat menggunakan InSpec bahasa Chef untuk membuat profil kustom Anda sendiri.

Jika server Anda belum menjalankan Chef Automate 2.0, Anda dapat mengatur Kepatuhan Chef secara manual, dengan menginstal buku masak Audit.

> **Note**
>
> Versi minimum yang didukung dari perangkat lunak agen klien Chef Infra (`chef-client`) pada node yang terkait dengan AWS OpsWorks for Chef Automate server adalah 13. x. Kami merekomendasikan menjalankan `chef-client` versi terbaru, stabil, atau setidaknya 14.10.9.

Topik

- Kepatuhan dalam Chef Automate 2.0
- Kepatuhan dalam Chef Automate 1. x
- Pembaruan Kepatuhan
- Profil Kepatuhan Komunitas dan Kustom
- Lihat Juga

Kepatuhan dalam Chef Automate 2.0

Jika AWS OpsWorks for Chef Automate server Anda menjalankan Chef Automate 2.0, siapkan Kepatuhan Chef dengan menggunakan prosedur di bagian ini.

Menjalankan Pekerjaan Pemindaian Kepatuhan dengan Chef Automate 2.0

Chef Automate 2.0 mencakup kemampuan InSpec pemindaian kepatuhan Chef yang sebelumnya memerlukan pengaturan manual dan konfigurasi buku masak. Anda dapat menjalankan pekerjaan pemindaian di AWS OpsWorks for Chef Automate server yang menjalankan Chef Automate 2.0. Pekerjaan dapat dijalankan segera (satu kali), dijadwalkan untuk lain waktu, atau dijadwalkan untuk berjalan pada interval tertentu, seperti setiap hari atau setiap dua jam. Hasil pekerjaan pemindaian dikirim ke pelaporan kepatuhan. Anda dapat melihat dan mengambil tindakan pada hasil pemindaian kepatuhan di dasbor Chef Automate. Untuk membuka tab Kepatuhan dan melihat laporan, pada tab Pindai Pekerjaan di dasbor Chef Automate, pilih Laporkan di sebelah kanan baris node terkelola.

Untuk menjalankan pekerjaan pemindaian pada node terkelola, Anda harus memiliki yang berikut ini.

- Setidaknya satu profil kepatuhan dipasang di namespace Anda.

- Setidaknya satu node target, baik yang ditambahkan secara manual, atau EC2 instance ditambahkan secara otomatis.

Di AWS OpsWorks for Chef Automate, pekerjaan pemindaian didukung pada target berikut.

- Node yang ditambahkan secara manual
- Instans `aws-ec2`
- Wilayah AWS

Untuk petunjuk mendetail tentang cara menjalankan pekerjaan pemindaian, lihat Chef Automate Scan Jobs di dokumentasi Chef.

(Opsional, Chef Automate 2.0) Menyiapkan Kepatuhan dengan Buku Masak Audit

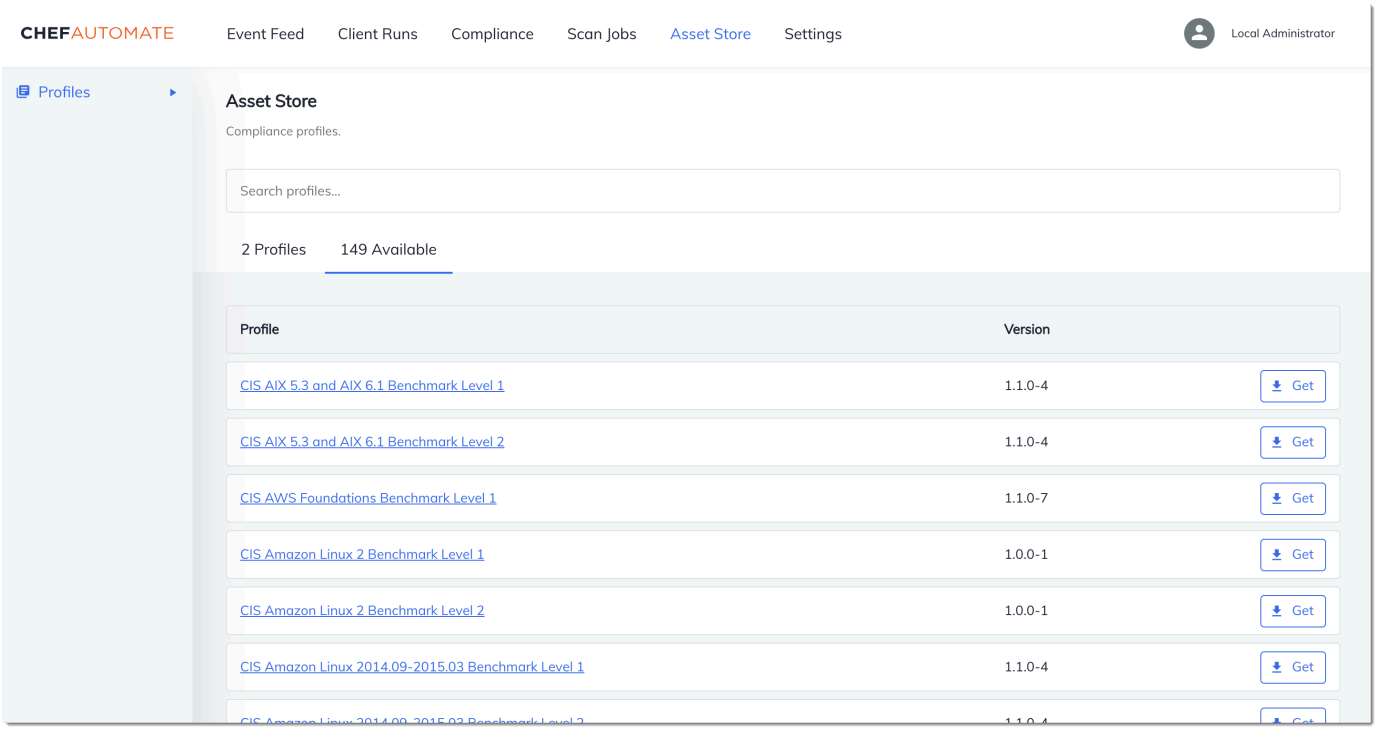
Anda dapat mengonfigurasi kepatuhan pada AWS OpsWorks for Chef Automate server mana pun. Setelah meluncurkan AWS OpsWorks for Chef Automate server, Anda dapat menginstal profil dari dasbor Chef Automate, atau menambahkan profil yang diinginkan ke atribut buku masak Audit di file `Policyfile.rb` kebijakan. `Policyfile.rb`File yang sudah diisi sebelumnya disertakan dalam starter kit.

Setelah Anda mengedit `Policyfile.rb` dengan profil sebagai atribut buku masak Audit, jalankan `chef push` perintah untuk mengunggah buku masak Audit dan buku masak lainnya yang ditentukan `Policyfile.rb` ke server Chef Automate Anda. Memasang buku masak Audit juga menginstal permata untuk Chef InSpec, kerangka pengujian dan audit sumber terbuka yang diproduksi oleh Chef. Untuk Chef Automate 2.0, pilih versi 7.1.0 atau yang lebih baru dari buku masak Audit. InSpec Permata harus versi 2.2.102 atau yang lebih baru.

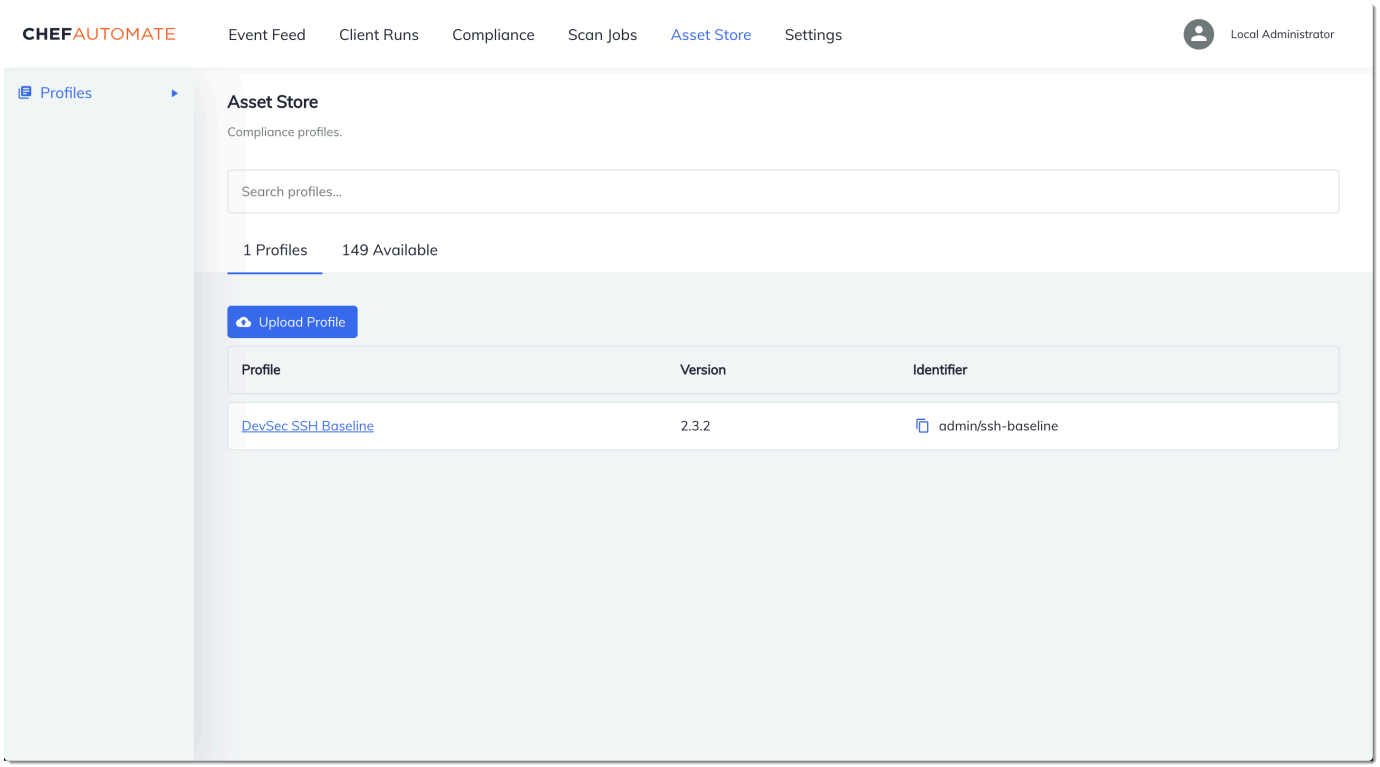
Petunjuk di bagian ini menunjukkan cara menerapkan `opsworks-audit` buku masak. Buku masak Audit mengunduh profil tertentu dari server Chef Automate, menilai node terhadap profil DevSec SSH Baseline, dan melaporkan hasil pemindaian kepatuhan pada setiap proses. `chef-client`

Untuk menginstal profil kepatuhan

1. Jika Anda belum melakukannya, masuk ke dasbor berbasis web Chef Automate. Gunakan kredenal yang Anda terima saat mengunduh Starter Kit saat Anda AWS OpsWorks for Chef Automate membuat server.
2. Di dasbor Chef Automate, pilih tab Asset Store.



3. Pilih tab Tersedia untuk melihat profil yang telah ditentukan.
4. Jelajahi daftar profil. Pilih profil yang cocok dengan sistem operasi dan konfigurasi setidaknya satu dari node terkelola Anda. Untuk melihat detail tentang profil, termasuk deskripsi pelanggaran yang ditargetkan profil dan kode aturan yang mendasarinya, pilih **>** di sebelah kanan entri profil. Anda dapat memilih beberapa profil. Jika Anda menyiapkan contoh di Starter Kit, pilih DevSec SSH Baseline.



5. Untuk menginstal profil yang dipilih di server Chef Automate Anda, pilih Dapatkan.
6. Setelah Anda menginstal profil, profil ditampilkan di tab Profil di dasbor Chef Automate.

Untuk menginstal buku masak dengan **`Policyfile.rb`**

1. Lihat `Policyfile.rb` di Starter Kit Anda untuk melihat bahwa atribut untuk buku masak Audit menentukan `ssh-baseline` profil di`['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. Unduh dan instal buku masak yang ditentukan dalam`Policyfile.rb`.

```
chef install
```

Semua buku masak diberi versi dalam file buku masak. `metadata.rb` Setiap kali Anda mengganti buku masak, Anda harus menaikkan versi buku masak yang ada di dalamnya. `metadata.rb`

3. Dorong kebijakan`opsworks-demo`, yang ditentukan dalam`Policyfile.rb`, ke server Anda.

```
chef push opsworks-demo
```

4. Verifikasi pemasangan kebijakan Anda. Jalankan perintah berikut.

```
chef show-policy
```

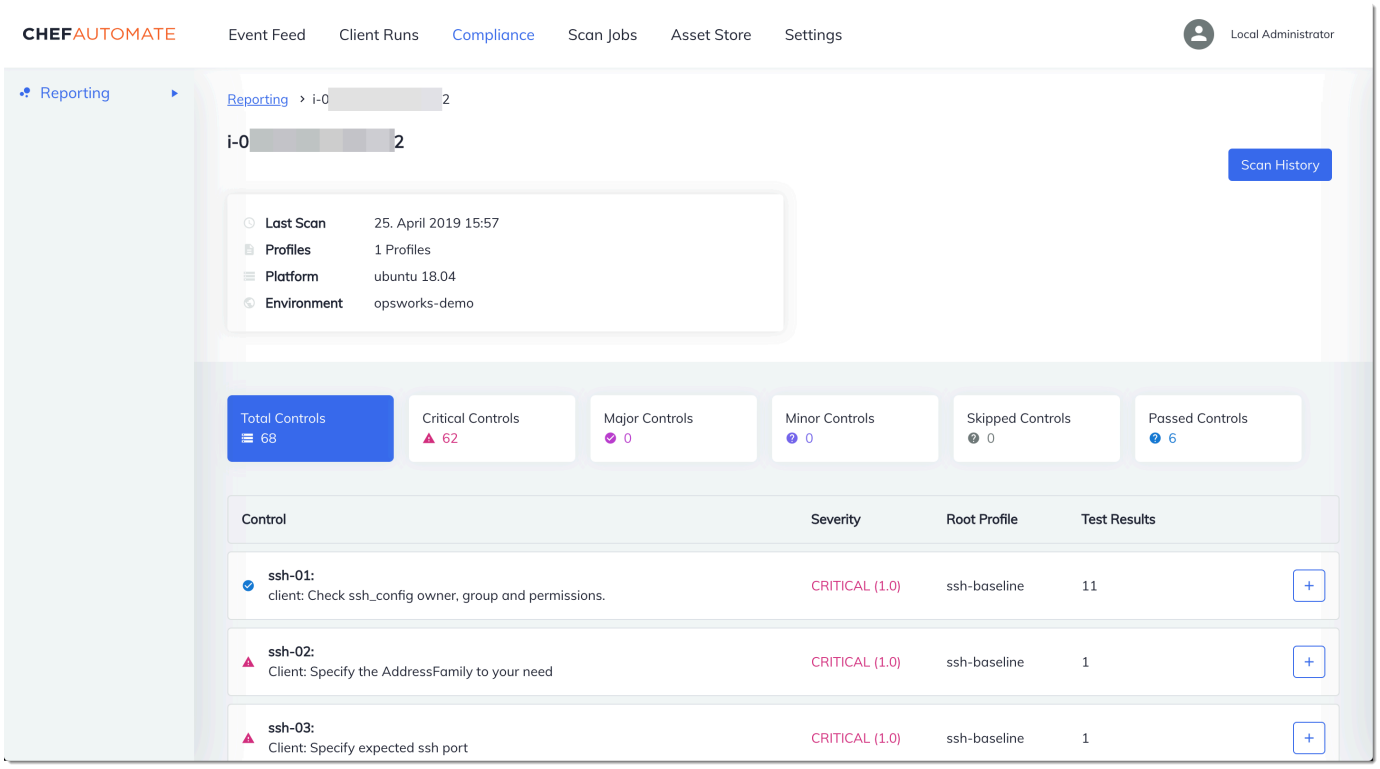
Hasilnya harus menyerupai yang berikut:

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

5. Tambahkan node ke server Anda untuk mengelola, jika Anda belum melakukannya. Untuk menghubungkan node pertama Anda ke AWS OpsWorks for Chef Automate server, gunakan `userdata.sh` skrip yang disertakan dalam Starter Kit ini. Ini menggunakan AWS OpsWorks `AssociateNode` API untuk menghubungkan node ke server Anda.

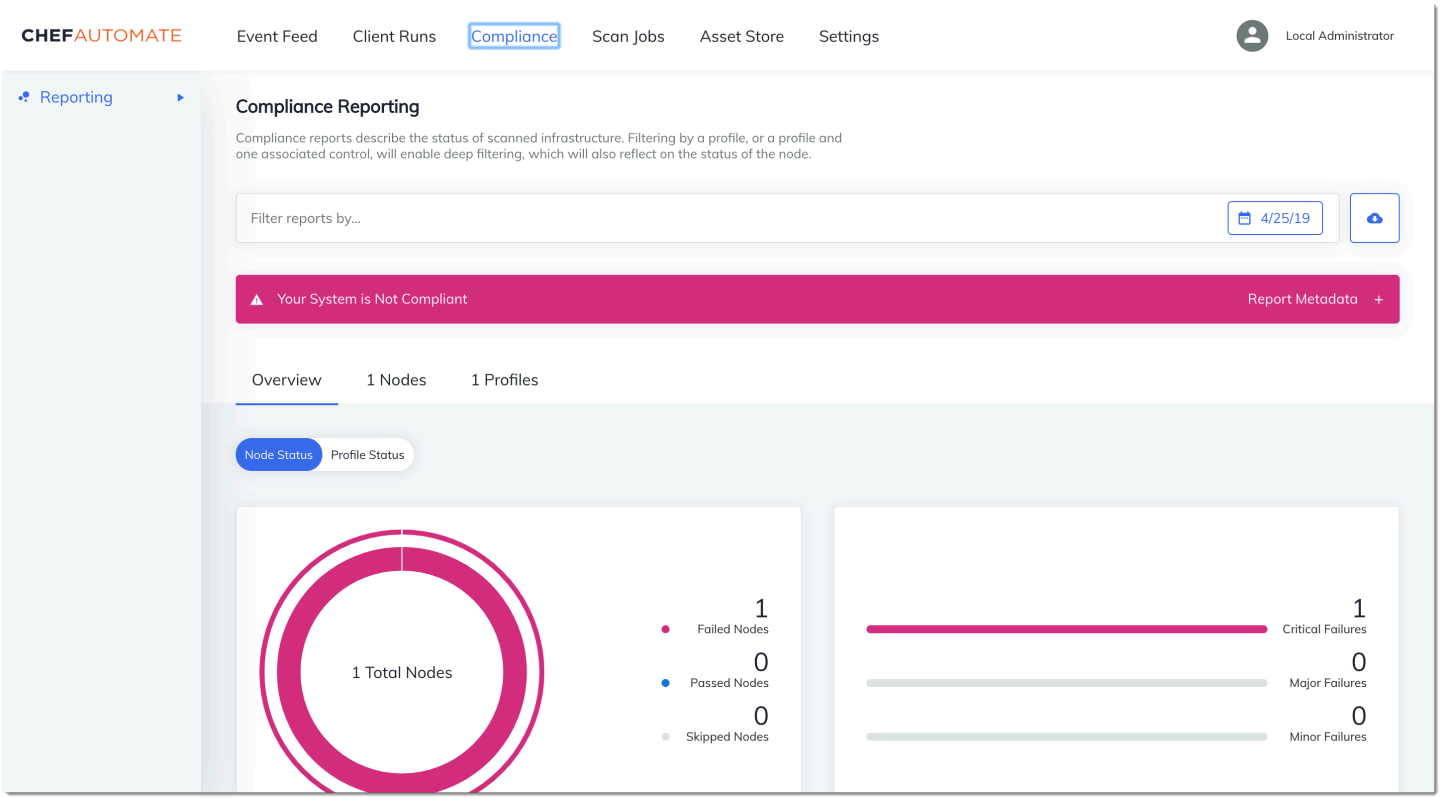
   Anda dapat mengotomatiskan asosiasi node dengan mengikuti langkah-langkah[Tambahkan node secara otomatis di AWS OpsWorks for Chef Automate](), atau menambahkan node satu per satu dengan mengikuti langkah-langkah dalam[Tambahkan node satu per satu]().

6. Setelah Anda memperbarui daftar run untuk node Anda, `chef-client` agen menjalankan resep yang Anda tentukan pada proses berikutnya. Secara default, ini terjadi setiap 1800 detik (30 menit). Setelah dijalankan, Anda dapat melihat dan mengambil tindakan pada hasil kepatuhan dari tab Kepatuhan di dasbor Chef Automate.

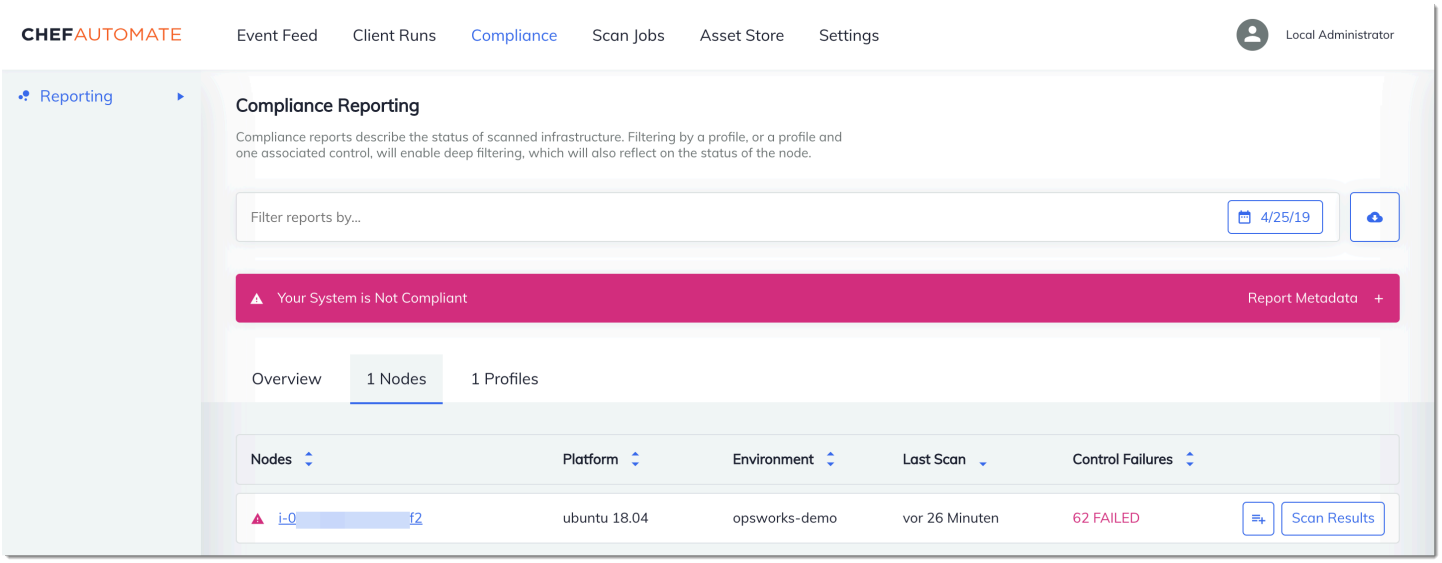


Menjalankan Pemindaian Kepatuhan

Anda akan melihat hasil pemindaian kepatuhan di dasbor Chef Automate segera setelah menjalankan agen pertama yang terjadi setelah Anda mengonfigurasi daftar run node.



Di dasbor Chef Automate, pilih tab Kepatuhan. Di panel navigasi kiri, pilih Pelaporan. Pilih tab Profil, pilih Hasil Pindai, lalu pilih simpul dengan kegagalan pemindaian untuk mempelajari lebih lanjut tentang aturan yang menyebabkan node gagal.



Biasanya, Anda melihat hasil pemindaian yang tidak sesuai, karena node baru belum memenuhi semua aturan di profil Baseline DevSec SSH. The DevSec Hardening Framework, sebuah proyek

berbasis komunitas, menawarkan buku masak untuk memperbaiki masalah yang melanggar aturan dalam profil SSH Baseline. DevSec

(Opsional) Menyelesaikan Hasil yang Tidak Sesuai

Starter kit menyertakan buku masak sumber terbuka,**`ssh-hardening`**, yang dapat Anda jalankan untuk memperbaiki hasil yang tidak sesuai dari proses terhadap profil SSH Baseline. DevSec

> **Note**
>
> `ssh-hardening`Buku masak membuat perubahan pada node Anda untuk mematuhi aturan DevSec SSH Baseline. Sebelum menjalankan buku masak ini di node produksi apa pun, tinjau detail tentang profil DevSec SSH Baseline di konsol Chef Automate untuk memahami pelanggaran aturan yang ditargetkan oleh buku masak. Tinjau informasi tentang `ssh-hardening`buku masak sumber terbuka sebelum menjalankannya di node produksi apa pun.

Untuk menjalankan buku **`ssh-hardening`** masak

1. Dalam editor teks, tambahkan `ssh-hardening` buku masak ke daftar run. `Policyfile.rb` Daftar `Policyfile.rb` run harus cocok dengan yang berikut ini.

   ```
   run_list  'chef-client', 'opsworks-webserver', 'audit', 'ssh-hardening'
   ```

2. Perbarui`Policyfile.rb`, dan dorong ke AWS OpsWorks for Chef Automate server Anda.

   ```
   chef update Policyfile.rb
      chef push opsworks-demo
   ```

3. Node yang terkait dengan `opsworks-demo` kebijakan memperbarui daftar jalankan secara otomatis, dan menerapkan `ssh-hardening` buku masak pada proses berikutnya`chef-client`.

   Karena Anda menggunakan `chef-client` buku masak, node Anda memeriksa secara berkala (secara default, setiap 30 menit). Pada check-in berikutnya, `ssh-hardening` buku masak berjalan, dan membantu meningkatkan keamanan node untuk memenuhi aturan profil DevSec SSH Baseline.

4. Setelah menjalankan awal `ssh-hardening` buku masak, tunggu 30 menit agar pemindaian kepatuhan berjalan lagi. Lihat hasilnya di dasbor Chef Automate. Hasil yang tidak sesuai yang terjadi pada awal pemindaian DevSec SSH Baseline harus diselesaikan.

Kepatuhan dalam Chef Automate 1. x

Jika AWS OpsWorks for Chef Automate server Anda menjalankan Chef Automate 1. x, atur Kepatuhan Koki dengan menggunakan prosedur di bagian ini.

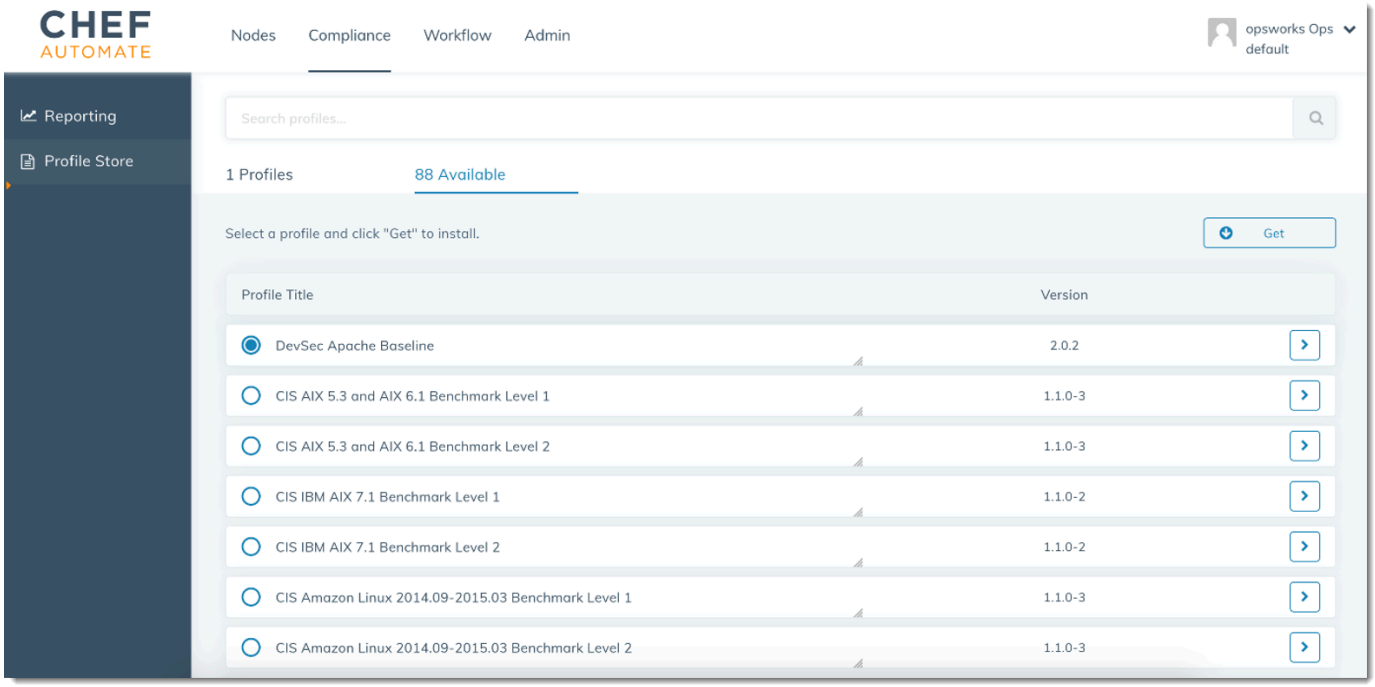
(Opsional, Chef Automate 1. x) Menyiapkan Kepatuhan Koki

Anda dapat mengonfigurasi Kepatuhan Koki di AWS OpsWorks for Chef Automate server mana pun. Setelah Anda meluncurkan AWS OpsWorks for Chef Automate server, pilih profil yang ingin Anda jalankan dari profil di dasbor Chef Automate. Setelah Anda menginstal profil, jalankan `berks` perintah untuk mengunggah buku masak Audit ke server Chef Automate Anda. Menginstal buku masak Audit juga menginstal permata untuk InSpec, kerangka pengujian sumber terbuka yang diproduksi oleh Chef yang memungkinkan Anda mengintegrasikan pengujian otomatis ke dalam setiap tahap pipeline penerapan Anda. Untuk Chef Automate 1. x, pilih versi 5.0.1 atau yang lebih baru dari buku masak Audit. InSpec Permata harus versi 1.24.0 atau yang lebih baru.

AWS OpsWorks for Chef Automate Starter kit mencakup buku masak pembungkus,`opsworks-audit`, yang mengunduh dan menginstal versi buku masak Chef's Audit yang tepat untuk Anda. `opsworks-audit`Buku masak juga menginstruksikan `chef-client` agen untuk menilai node terhadap profil DevSecSSH Baseline yang Anda instal dari konsol Kepatuhan Chef nanti dalam topik ini. Anda dapat mengatur Kepatuhan dengan menggunakan salah satu buku masak yang sesuai dengan preferensi Anda. Petunjuk di bagian ini menunjukkan cara menerapkan `opsworks-audit` buku masak.

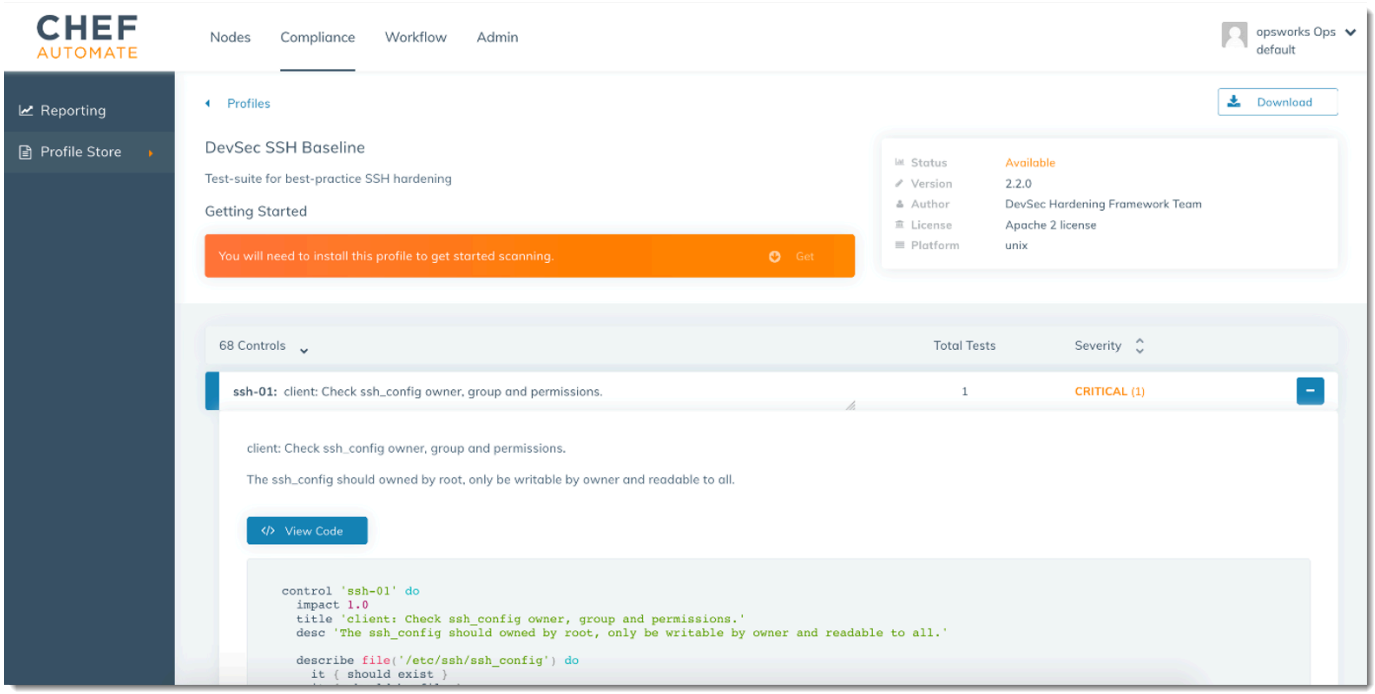
Untuk menginstal profil Kepatuhan

1. Jika Anda belum melakukannya, masuk ke dasbor berbasis web Chef Automate. Gunakan kredenal yang Anda terima saat mengunduh Starter Kit saat Anda AWS OpsWorks for Chef Automate membuat server.
2. Di dasbor Chef Automate, pilih tab Kepatuhan.



3. Di bilah navigasi kiri, pilih Toko Profil, lalu pilih tab Tersedia untuk melihat profil yang telah ditentukan sebelumnya.

4. Jelajahi daftar profil. Pilih profil yang cocok dengan sistem operasi dan konfigurasi setidaknya satu dari node terkelola Anda. Untuk melihat detail tentang profil, termasuk deskripsi pelanggaran yang ditargetkan profil dan kode aturan yang mendasarinya, pilih **>** di sebelah kanan entri profil. Anda dapat memilih beberapa profil.



5. Untuk menginstal profil yang dipilih di server Chef Automate Anda, pilih Dapatkan.
6. Ketika unduhan selesai, lanjutkan ke prosedur selanjutnya.

Untuk menginstal dan mengatur buku **`opsworks-audit`** masak

1. Langkah ini opsional, tetapi menghemat waktu di Langkah 6, saat Anda menambahkan resep ke daftar run node. Edit `roles/opsworks-example-role.rb` file yang disertakan dalam starter kit yang Anda unduh selama pembuatan AWS OpsWorks for Chef Automate server Anda. Tambahkan baris berikut. Baris terakhir dikomentari, karena menambahkan `ssh-hardening` buku masak dan resep untuk menyelesaikan node yang tidak sesuai setelah pemindaian Kepatuhan Anda berjalan adalah opsional.

   ```
   run_list(
        "recipe[chef-client]",
        "recipe[apache2]",
        "recipe[opsworks-audit]"
        # "recipe[ssh-hardening]"
          )
   ```

2. Gunakan editor teks untuk menentukan buku masak yang Anda inginkan di Berksfile Anda. Sampel Berksfile disediakan untuk Anda di starter kit. Dalam contoh ini, kami menginstal buku masak Chef Infra client (`chef-client`), buku `apache2` masak, dan buku masak. `opsworks-audit` Berksfile Anda harus menyerupai yang berikut ini.

   ```
   source 'https://supermarket.chef.io
        cookbook 'chef-client'
        cookbook 'apache2', '~> 5.0.1'
        cookbook 'opsworks-audit', path: 'cookbooks/opsworks-audit', '~> 1.0.0'
   ```

   Semua buku masak diberi versi dalam file buku masak. `metadata.rb` Setiap kali Anda mengganti buku masak, Anda harus menaikkan versi buku masak yang ada di dalamnya. `metadata.rb`

3. Jalankan perintah berikut untuk mengunduh dan menginstal buku masak ke `cookbooks` folder di komputer lokal atau komputer Anda yang berfungsi.

   ```
   berks vendor cookbooks
   ```

4. Jalankan perintah berikut untuk mengunggah buku masak vendor ke server Anda. AWS OpsWorks for Chef Automate

```
knife upload .
```

5. Jalankan perintah berikut untuk memverifikasi instalasi `opsworks-audit` buku masak dengan menunjukkan daftar buku masak yang saat ini tersedia di server.

```
knife cookbook list
```

6. Tambahkan node ke server Anda untuk mengelola, jika Anda belum melakukannya. Anda dapat mengotomatiskan asosiasi node dengan mengikuti langkah-langkahTambahkan node secara otomatis di AWS OpsWorks for Chef Automate, atau menambahkan node satu per satu dengan mengikuti langkah-langkah dalamTambahkan node satu per satu. Edit daftar run node Anda untuk menambahkan peran yang Anda tentukan di Langkah 1,`opsworks-example-role`. Dalam contoh ini, kami mengedit `RUN_LIST` atribut dalam `userdata` skrip yang Anda gunakan untuk mengotomatiskan asosiasi node.

```
RUN_LIST="role[opsworks-example-role]"
```

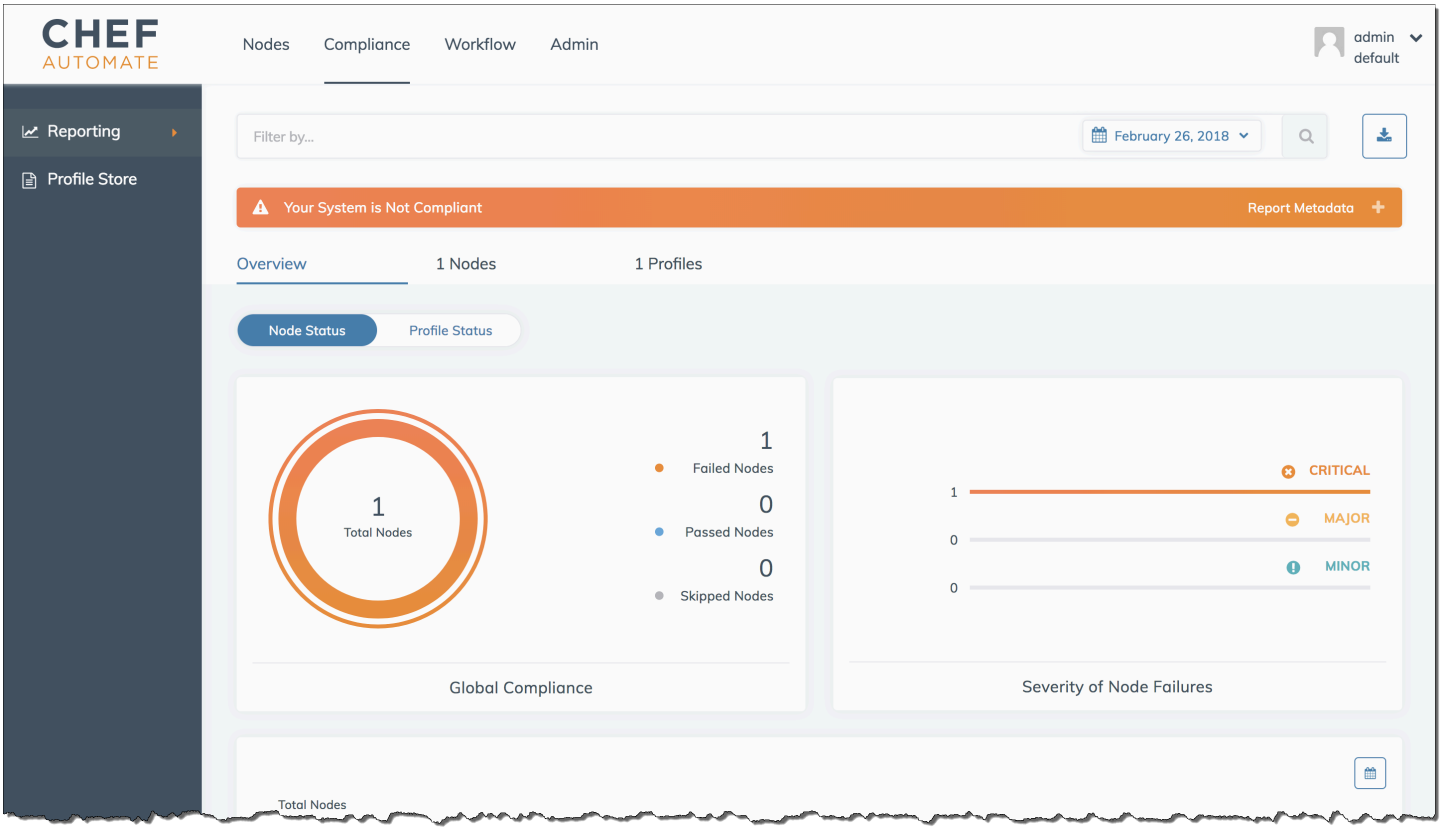
   Jika Anda melewati Langkah 1, dan tidak mengatur peran, tambahkan nama masing-masing resep ke daftar run. Simpan perubahan Anda, dan ikuti langkah-langkah Langkah 3: Buat Instans dengan Menggunakan Skrip Asosiasi Tanpa Pengawasan untuk menerapkan skrip data pengguna Anda ke EC2 instans Amazon.

```
RUN_LIST="recipe[chef-client],recipe[apache2],recipe[opworks-audit]"
```

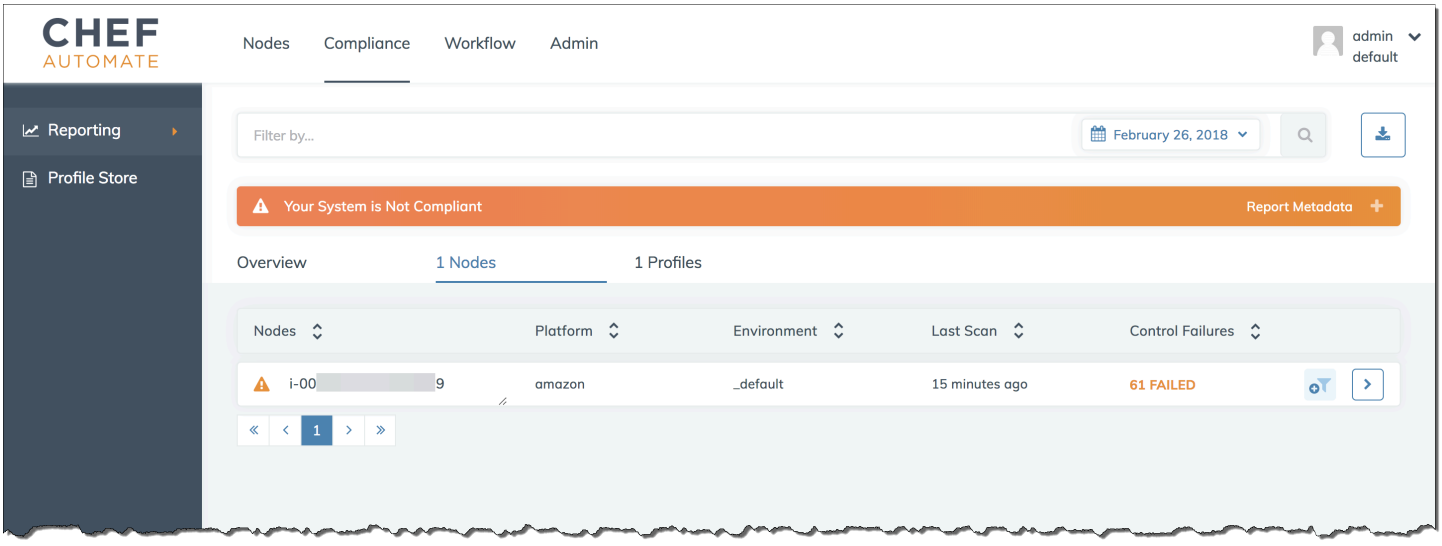
7. Setelah Anda memperbarui daftar run untuk node Anda, `chef-client` agen menjalankan resep yang Anda tentukan pada proses berikutnya. Secara default, ini terjadi setiap 1800 detik (30 menit). Setelah dijalankan, hasil Kepatuhan Anda akan terlihat di dasbor Chef Automate.

Menjalankan Pemindaian Kepatuhan

Anda akan melihat hasil pemindaian kepatuhan di dasbor Chef Automate segera setelah menjalankan pertama daemon agen yang terjadi setelah Anda mengonfigurasi daftar run node.



Di dasbor Chef Automate, pilih tab Kepatuhan. Di panel navigasi kiri, pilih Pelaporan. Pilih tab Profil, pilih Hasil Pindai, lalu pilih simpul dengan kegagalan pemindaian untuk mempelajari lebih lanjut tentang aturan yang menyebabkan node gagal.



Biasanya, Anda melihat hasil pemindaian yang tidak sesuai, karena node baru belum memenuhi semua aturan di profil Baseline DevSec SSH. The DevSec Hardening Framework, sebuah proyek

berbasis komunitas, menawarkan buku masak untuk memperbaiki masalah yang melanggar aturan dalam profil SSH Baseline. DevSec

(Opsional) Menyelesaikan Hasil yang Tidak Sesuai

Starter kit menyertakan buku masak sumber terbuka,**`ssh-hardening`**, yang dapat Anda jalankan untuk memperbaiki hasil yang tidak sesuai dari proses terhadap profil SSH Baseline. DevSec

> **Note**
>
> `ssh-hardening`Buku masak membuat perubahan pada node Anda untuk mematuhi aturan DevSec SSH Baseline. Sebelum menjalankan buku masak ini di node produksi apa pun, tinjau detail tentang profil DevSec SSH Baseline di konsol Chef Automate untuk memahami pelanggaran aturan yang ditargetkan oleh buku masak. Tinjau informasi tentang `ssh-hardening`buku masak sumber terbuka sebelum menjalankannya di node produksi apa pun.

Untuk menjalankan buku **`ssh-hardening`** masak

1. Dalam editor teks, tambahkan `ssh-hardening` buku masak ke Berksfile Anda. Berksfile Anda harus menyerupai yang berikut ini.

   ```
   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. Jalankan perintah berikut untuk mengunduh `ssh-hardening` buku masak ke folder buku masak lokal Anda, lalu unggah ke server Anda AWS OpsWorks for Chef Automate .

   ```
   berks vendor cookbooks
   knife upload .
   ```

3. Tambahkan `ssh-hardening` resep ke daftar run node Anda seperti yang dijelaskan dalam Langkah 1 dan 6Untuk menginstal dan mengatur buku `opsworks-audit` masak.

   Jika Anda memperbarui `opsworks-example-role.rb` file, unggah perubahan Anda ke server Anda dengan menjalankan perintah berikut.

```
knife upload .
```

Jika Anda memperbarui daftar run secara langsung, unggah perubahan dengan menjalankan perintah berikut. Nama node biasanya ID instance.

```
knife node run_list add <node name> 'recipe[ssh-hardening]'
```

4. Karena Anda menggunakan `chef-client` buku masak, node Anda memeriksa secara berkala (secara default, setiap 30 menit). Pada check-in berikutnya, `ssh-hardening` buku masak berjalan, dan membantu meningkatkan keamanan node untuk memenuhi aturan profil DevSec SSH Baseline.
5. Setelah menjalankan awal `ssh-hardening` buku masak, tunggu 30 menit agar pemindaian Kepatuhan berjalan kembali. Lihat hasilnya di dasbor Chef Automate. Hasil yang tidak sesuai yang terjadi pada awal pemindaian DevSec SSH Baseline harus diselesaikan.

Pembaruan Kepatuhan

Di AWS OpsWorks for Chef Automate server, fungsionalitas kepatuhan diperbarui secara otomatis oleh pemeliharaan sistem terjadwal Anda. Saat rilis terbaru Chef Automate, Chef Infra Server, dan Chef InSpec tersedia untuk AWS OpsWorks for Chef Automate server Anda, Anda mungkin perlu memeriksa dan memperbarui versi buku masak Audit dan InSpec permata Chef yang didukung yang berjalan di server Anda. Profil yang telah Anda instal di AWS OpsWorks for Chef Automate server Anda tidak diperbarui sebagai bagian dari pemeliharaan.

Profil Kepatuhan Komunitas dan Kustom

Chef saat ini mencakup lebih dari 100 profil pemindaian kepatuhan. Anda dapat menambahkan profil komunitas dan kustom ke daftar, lalu mengunduh dan menjalankan pemindaian kepatuhan berdasarkan profil tersebut, seperti yang Anda lakukan untuk profil yang disertakan. Profil kepatuhan berbasis komunitas tersedia dari Chef Supermarket. Profil kustom adalah program berbasis Ruby yang menyertakan folder kontrol yang menentukan aturan pemindaian Anda.

Lihat Juga

- Posting blog pengumuman Kepatuhan Chef
- Pelatihan online Kepatuhan Chef Automate

- InSpecSitus Chef
- InSpec Tutorial Koki

Memutuskan Node dari Server AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara memisahkan, atau menghapus, node terkelola dari manajemen oleh AWS OpsWorks for Chef Automate server. Operasi ini dilakukan pada baris perintah; Anda tidak dapat memisahkan node di konsol AWS OpsWorks for Chef Automate manajemen. Saat ini, AWS OpsWorks for Chef Automate API tidak mengizinkan penghapusan batch beberapa node. Perintah di bagian ini memisahkan satu node pada satu waktu.

Kami menyarankan Anda memisahkan node dari server Chef sebelum Anda menghapus server, sehingga node terus beroperasi tanpa mencoba untuk terhubung kembali dengan server. Untuk melakukan ini, jalankan `disassociate-node` AWS CLI perintah.

Untuk memisahkan node

1. Dalam AWS CLI, jalankan perintah berikut untuk memisahkan node. *`Node_name`*adalah nama node yang ingin Anda lepaskan; untuk EC2 instance Amazon, ini adalah ID instance. *`Server_name`*adalah nama server Chef dari mana Anda ingin memisahkan node. `--engine-attributes`menentukan `CHEF_AUTOMATE_ORGANIZATION` nama default Anda. Ketiga parameter ini diperlukan.

   `--region`Parameter tidak diperlukan kecuali Anda ingin memisahkan node dari server Chef yang tidak ada di wilayah default Anda.

```
aws opsworks-cm --region Region_name disassociate-node --node-
name Node_name --server-name Server_name --engine-attributes
 "Name=CHEF_AUTOMATE_ORGANIZATION,Value='default'"
```

Berikut adalah contoh perintah tersebut.

```
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. Tunggu sampai pesan respons menunjukkan bahwa disasosiasi selesai.

   Setelah Anda berhasil memisahkan node dari AWS OpsWorks for Chef Automate server, itu mungkin masih terlihat di dasbor Chef Automate. Secara default, Chef memberlakukan periode retensi untuk informasi status node, dan membersihkan node secara otomatis setelah beberapa hari.

Untuk informasi selengkapnya tentang cara menghapus AWS OpsWorks for Chef Automate server, lihatMenghapus AWS OpsWorks for Chef Automate Server.

Topik Terkait

Posting AWS blog berikut menawarkan informasi lebih lanjut tentang menghubungkan node secara otomatis dengan server Chef Automate Anda, dengan menggunakan grup Auto Scaling, atau dalam beberapa akun.

- Menggunakan AWS OpsWorks for Chef Automate untuk Mengelola EC2 Instans dengan Auto Scaling
- OpsWorks untuk Chef Automate - Secara Otomatis Bootstrapping Node di Akun yang Berbeda

Menghapus AWS OpsWorks for Chef Automate Server

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan,

Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara menghapus AWS OpsWorks for Chef Automate server. Menghapus server juga menghapus peristiwa, log, dan buku masak apa pun yang disimpan di server. Sumber daya pendukung (instans Amazon Elastic Compute Cloud, volume Amazon Elastic Block Store, dll.) juga dihapus, bersama dengan semua pencadangan otomatis.

Meskipun menghapus server tidak menghapus node, mereka tidak lagi dikelola oleh server yang dihapus, dan akan terus mencoba untuk menyambung kembali. Untuk alasan ini, kami sarankan untuk memisahkan node terkelola sebelum Anda menghapus server Chef. Dalam rilis ini, Anda dapat memisahkan node dengan menjalankan AWS CLI perintah.

Langkah 1: Pisahkan Node yang Dikelola

Putuskan node dari server Chef sebelum Anda menghapus server, sehingga node terus beroperasi tanpa mencoba untuk terhubung kembali dengan server. Untuk melakukan ini, jalankan `disassociate-node` AWS CLI perintah.

Untuk memisahkan node

1. Dalam AWS CLI, jalankan perintah berikut untuk memisahkan node. *`Server_name`*adalah nama server Chef dari mana Anda ingin memisahkan node.

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

2. Tunggu sampai pesan respons menunjukkan bahwa disasosiasi selesai.

Langkah 2: Hapus Server

1. Pada ubin server di dasbor, perluas menu Tindakan.
2. Pilih Hapus server.
3. Ketika Anda diminta untuk mengonfirmasi penghapusan, pilih Ya.

Atur Ulang Kredensi Dasbor Otomatis Chef

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Secara berkala, Anda mungkin ingin mengubah kata sandi yang digunakan untuk masuk ke dasbor Chef Automate. Anda juga dapat menggunakan AWS CLI perintah Amazon EC2 Systems Manager yang ditampilkan di bagian ini untuk mengubah kata sandi dasbor Chef Automate jika Anda kehilangannya. Perintah yang Anda gunakan tergantung pada apakah server Chef Automate Anda menjalankan versi 1 atau versi 2 dari Chef Automate.

1. Untuk mengembalikan ID instance server Chef Anda, buka AWS Management Console ke halaman berikut.

   https://console.aws.amazon.com/ec2/v2/rumah? wilayah= #Contoh: pencarian = *region_of_your_server* - aws-opsworks-cm *server_name*

   Misalnya, untuk server Chef bernama MyChefServerdi Wilayah Barat AS (Oregon), URL konsol adalah sebagai berikut.

   https://console.aws.amazon.com/ec2/v2/rumah? region=us-barat-2 #instances:search= - aws-opsworks-cm MyChefServer

   Catat ID instance yang ditampilkan di konsol; Anda akan memerlukannya untuk mengubah kata sandi Anda.
2. Untuk mengatur ulang kata sandi masuk dasbor Chef Automate, jalankan salah satu AWS CLI perintah berikut, tergantung pada apakah server Anda menjalankan Chef Automate 1 atau Chef Automate 2. Ganti *enterprise_name* dengan nama perusahaan atau organisasi Anda, *user_name* dengan nama pengguna administrator di server, *new_password* dengan kata sandi yang ingin Anda gunakan, dan *region_name* dengan wilayah tempat server Anda berada. Jika Anda tidak menentukan nama perusahaan, nama perusahaan akan menjadi`default`. Secara default, *enterprise_name* is `default` (ini adalah nama organisasi yang selalu

disediakan). Untuk*user_name*, AWS OpsWorks for Chef Automate hanya membuat pengguna bernama`admin`. Catat kata sandi baru, dan simpan di lokasi yang aman namun nyaman.

Untuk 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
```

Untuk 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. Tunggu teks keluaran (dalam hal ini, ID perintah) untuk menunjukkan bahwa perubahan kata sandi selesai.

Pencatatan Panggilan AWS OpsWorks for Chef Automate API dengan AWS CloudTrail

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks for Chef Automate terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh identitas IAM, atau AWS layanan di AWS OpsWorks for Chef Automate. CloudTrail menangkap semua panggilan API untuk AWS OpsWorks for Chef Automate sebagai peristiwa, termasuk panggilan dari AWS OpsWorks for Chef Automate konsol dan dari panggilan kode ke AWS OpsWorks for Chef Automate APIs. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk

acara untuk. AWS OpsWorks for Chef Automate Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat AWS OpsWorks for Chef Automate, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

AWS OpsWorks for Chef Automate Informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di AWS OpsWorks for Chef Automate, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk AWS OpsWorks for Chef Automate, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, ketika Anda membuat jejak di konsol, jejak ini diterapkan ke semua Wilayah. Jejak mencatat peristiwa dari semua wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

Semua AWS OpsWorks for Chef Automate tindakan dicatat oleh CloudTrail dan didokumentasikan dalam Referensi AWS OpsWorks for Chef Automate API. Misalnya, panggilan ke CreateServer, CreateBackup, dan DescribeServerstindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan tersebut dibuat dengan kredensial root atau pengguna IAM.

- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

Memahami Entri File AWS OpsWorks for Chef Automate Log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log untuk AWS OpsWorks for Chef Automate `CreateServer` tindakan tersebut.

```
{"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"
}
```


Pemecahan masalah AWS OpsWorks for Chef Automate

> **Important**
>
> AWS OpsWorks untuk Chef Automate mencapai akhir masa pakai pada 5 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan lama. Kami menyarankan agar pelanggan yang sudah ada bermigrasi ke Chef SaaS atau solusi alternatif. Jika Anda memiliki pertanyaan, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Topik ini berisi beberapa AWS OpsWorks for Chef Automate masalah umum, dan solusi yang disarankan untuk masalah tersebut.

Topik

- Kiat pemecahan masalah umum
- Memecahkan masalah kesalahan tertentu
- Bantuan dan dukungan tambahan

Kiat pemecahan masalah umum

Jika Anda tidak dapat membuat atau bekerja dengan server Chef, Anda dapat melihat pesan kesalahan atau log untuk membantu Anda memecahkan masalah. Tugas-tugas berikut menjelaskan tempat umum untuk memulai ketika Anda memecahkan masalah server Chef. Untuk informasi tentang kesalahan dan solusi tertentu, lihat Memecahkan masalah kesalahan tertentu bagian dari topik ini.

- Gunakan AWS OpsWorks for Chef Automate konsol untuk melihat pesan kesalahan jika server Chef gagal memulai. Pada halaman detail server Chef, pesan kesalahan yang terkait dengan peluncuran dan menjalankan server ditampilkan di bagian atas halaman. Kesalahan dapat berasal dari AWS OpsWorks for Chef Automate, AWS CloudFormation, atau Amazon EC2, layanan yang

digunakan untuk membuat server Chef. Pada halaman detail, Anda juga dapat melihat peristiwa yang terjadi di server yang sedang berjalan, yang dapat berisi pesan peristiwa kegagalan.

- Untuk membantu menyelesaikan EC2 masalah, sambungkan ke instance server Anda dengan menggunakan SSH, dan lihat log. EC2 log contoh disimpan dalam `/var/log/aws/opsworks-cm` direktori. Log ini menangkap output perintah saat AWS OpsWorks for Chef Automate meluncurkan server Chef.

Memecahkan masalah kesalahan tertentu

Topik

- Server dalam keadaan koneksi hilang
- Node terkelola muncul di dasbor Chef Automate di kolom Missing
- Tidak dapat membuat lemari besi Chef; knife vault perintah gagal dengan kesalahan
- Pembuatan server gagal dengan pesan “konfigurasi yang diminta saat ini tidak didukung”
- Server Chef tidak mengenali nama organisasi yang ditambahkan di dasbor Chef Automate
- Tidak dapat membuat EC2 instance Amazon server
- Kesalahan peran layanan mencegah pembuatan server
- Batas alamat IP elastis terlampaui
- Tidak dapat masuk ke dasbor Chef Automate
- Asosiasi simpul tanpa pengawasan gagal
- Pemeliharaan sistem gagal

Server dalam keadaan koneksi hilang

Masalah: Status server ditampilkan sebagai Koneksi hilang.

Penyebab: Ini paling sering terjadi ketika entitas di luar AWS OpsWorks membuat perubahan pada AWS OpsWorks for Chef Automate server atau sumber daya pendukungnya. AWS OpsWorks tidak dapat terhubung ke server Chef Automate di status yang hilang Koneksi untuk menangani tugas pemeliharaan seperti membuat cadangan, menerapkan tambalan sistem operasi, atau memperbarui Chef Automate. Akibatnya, server Anda mungkin kehilangan pembaruan penting, rentan terhadap masalah keamanan, atau tidak beroperasi seperti yang diharapkan.

Solusi: Coba langkah-langkah berikut untuk memulihkan koneksi server.

1. Pastikan bahwa peran layanan Anda memiliki semua izin yang diperlukan.

   a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk peran layanan yang digunakan server. Ini membuka peran layanan untuk dilihat di konsol IAM.

   b. Pada tab Izin, verifikasi yang `AWSOpsWorksCMServiceRole` ada di daftar Kebijakan izin. Jika tidak terdaftar, tambahkan kebijakan `AWSOpsWorksCMServiceRole` terkelola secara manual ke peran.

   c. Pada tab Hubungan kepercayaan, verifikasi bahwa peran layanan memiliki kebijakan kepercayaan yang mempercayai `opsworks-cm.amazonaws.com` layanan untuk mengambil peran atas nama Anda. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

2. Pastikan bahwa profil instans Anda memiliki semua izin yang diperlukan.

   a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk profil instance yang digunakan server. Ini membuka profil instance untuk dilihat di konsol IAM.

   b. Pada tab Izin, verifikasi itu `AmazonEC2RoleforSSM` dan keduanya `AWSOpsWorksCMInstanceProfileRole` ada di daftar Kebijakan Izin. Jika salah satu atau keduanya tidak terdaftar, tambahkan kebijakan terkelola ini secara manual ke peran.

   c. Pada tab Hubungan kepercayaan, verifikasi bahwa peran layanan memiliki kebijakan kepercayaan yang mempercayai `ec2.amazonaws.com` layanan untuk mengambil peran atas nama Anda. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

3. Di EC2 konsol Amazon, pastikan Anda berada di wilayah yang sama dengan wilayah AWS OpsWorks for Chef Automate server, lalu restart EC2 instance yang digunakan server Anda.

   a. Pilih EC2 contoh yang diberi nama `aws-opsworks-cm-instance-`*`server-name`*.

   b. Pada menu status Instance, pilih Reboot instance.

   c. Biarkan hingga 15 menit agar server Anda restart dan sepenuhnya online.

4. Di AWS OpsWorks for Chef Automate konsol, pada halaman detail server, verifikasi bahwa status server sekarang sehat.

Jika status server masih Koneksi hilang setelah melakukan langkah-langkah sebelumnya, coba salah satu dari berikut ini.

- Ganti server dengan membuat yang baru dan menghapus yang asli. Jika data di server saat ini penting bagi Anda, pulihkan server dari cadangan terbaru, dan verifikasi data yang mutakhir sebelum menghapus server asli yang tidak responsif.
- Hubungi AWS dukungan.

Node terkelola muncul di dasbor Chef Automate di kolom Missing

Masalah: Node terkelola muncul di kolom Missing dasbor Chef Automate.

Penyebab: Ketika node tidak terhubung ke server Chef Automate selama lebih dari 12 jam, dan `chef-client` tidak dapat berjalan di node, node berubah dari statusnya sebelum periode 12 jam, dan pindah ke kolom Hilang dari dasbor Chef Automate.

Solusi: Verifikasi bahwa node sedang online. Coba jalankan `knife node show` *`node_name`* `--run-list` untuk melihat `chef-client` apakah dapat berjalan pada node, atau `knife node show -l` *`node_name`* untuk menampilkan semua informasi tentang node. Node mungkin offline atau terputus dari jaringan.

Tidak dapat membuat lemari besi Chef; **`knife vault`** perintah gagal dengan kesalahan

Masalah: Anda mencoba membuat vault di server Chef Automate Anda (seperti brankas untuk menyimpan kredensi untuk node berbasis Windows yang bergabung dengan domain) dengan menjalankan perintah. `knife vault` Perintah mengembalikan pesan kesalahan yang mirip dengan berikut ini.

```
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!
```

Pengguna penting tidak dikembalikan ketika Anda menjalankan `knife user list` dari jarak jauh, tetapi Anda dapat melihat pengguna penting dalam hasil ketika Anda menjalankan `chef-server-ctl user-show` perintah secara lokal di server Chef Automate Anda. Dengan kata lain, `knife vault` perintah Anda tidak dapat menemukan pengguna penting, tetapi Anda tahu itu ada.

Penyebab: Meskipun pengguna penting dianggap sebagai pengguna super di Chef, dan memiliki izin penuh, itu bukan anggota organisasi mana pun, termasuk `default` organisasi yang digunakan. AWS OpsWorks for Chef Automate Perintah `knife user list` mengembalikan semua pengguna yang ada di organisasi saat ini dalam konfigurasi Chef Anda. `chef-server-ctl user-show`Perintah mengembalikan semua pengguna terlepas dari organisasi, termasuk pengguna penting.

Solusi: Untuk memperbaiki masalah, tambahkan pengguna penting ke organisasi default dengan menjalankan. `knife opc`

Pertama, Anda harus menginstal plugin knife-opc.

```
chef gem install knife-opc
```

Setelah Anda menginstal plugin, jalankan perintah berikut untuk menambahkan pengguna penting ke organisasi default.

```
knife opc org user add default pivotal
```

Anda dapat memverifikasi bahwa pengguna penting adalah bagian dari organisasi default dengan menjalankan lagi. `knife user list` `pivotal`harus tercantum dalam hasil. Kemudian, coba lari `knife vault` lagi.

Pembuatan server gagal dengan pesan “konfigurasi yang diminta saat ini tidak didukung”

Masalah: Anda mencoba membuat server Chef Automate, tetapi pembuatan server gagal dengan pesan kesalahan yang mirip dengan “Konfigurasi yang diminta saat ini tidak didukung. Silakan periksa dokumentasi untuk konfigurasi yang didukung.”

Penyebab: Jenis instans yang tidak didukung mungkin telah ditentukan untuk server Chef Automate. Jika Anda memilih untuk membuat server Chef Automate di VPC yang memiliki penyewaan non-default, seperti server untuk instance khusus, semua instance di dalam VPC yang ditentukan juga harus memiliki penyewaan khusus atau host. Karena beberapa jenis instance, seperti t2, hanya

tersedia dengan penyewaan default, jenis instance server Chef Automate mungkin tidak didukung oleh VPC yang ditentukan, dan pembuatan server gagal.

Solusi: Jika Anda memilih VPC yang memiliki tenancy non-default, gunakan tipe instans m4, yang dapat mendukung penyewaan khusus.

Server Chef tidak mengenali nama organisasi yang ditambahkan di dasbor Chef Automate

Masalah: Anda telah menambahkan nama organisasi Alur Kerja baru di dasbor Chef Automate, atau menetapkan `CHEF_AUTOMATE_ORGANIZATION` nilai selain `"default"` dalam skrip asosiasi node tanpa pengawasan, tetapi asosiasi node gagal. AWS OpsWorks for Chef Automate Server Anda tidak mengenali nama organisasi baru.

Penyebab: Nama organisasi alur kerja dan nama organisasi server Chef tidak sama. Anda dapat membuat organisasi Alur Kerja baru di dasbor Chef Automate berbasis web, tetapi bukan nama organisasi server Chef. Anda dapat menggunakan dasbor Chef Automate hanya untuk melihat organisasi server Chef yang ada. Organisasi baru yang Anda buat di dasbor Chef Automate adalah organisasi Workflow, dan tidak dikenali oleh server Chef. Anda tidak dapat membuat nama organisasi baru dengan menentukannya dalam skrip asosiasi simpul. Mengacu pada nama organisasi dalam skrip asosiasi node ketika organisasi belum pertama kali ditambahkan ke server Chef akan menyebabkan asosiasi node gagal.

Solusi: Untuk membuat organisasi baru yang dikenali di server Chef, gunakan `knife opc org create`perintah, atau jalankan `chef-server-ctl org-create`.

Tidak dapat membuat EC2 instance Amazon server

Masalah: Pembuatan server gagal dengan pesan kesalahan yang mirip dengan berikut ini: “Sumber daya berikut gagal dibuat: [EC2Instance]. Gagal menerima 1 sinyal sumber daya dalam durasi yang ditentukan.

Penyebab: Ini kemungkinan besar karena EC2 instance tidak memiliki akses jaringan.

Solusi: Pastikan instans memiliki akses Internet keluar, dan agen AWS layanan dapat mengeluarkan perintah. Pastikan VPC Anda (VPC dengan subnet publik tunggal) mengaktifkan resolusi DNS, dan subnet Anda mengaktifkan pengaturan Auto-assign Public IP.

Kesalahan peran layanan mencegah pembuatan server

Masalah: Pembuatan server gagal dengan pesan kesalahan yang menyatakan, “Tidak diizinkan untuk melakukan sts:AssumeRole.”

Penyebab: Ini dapat terjadi ketika peran layanan yang Anda gunakan tidak memiliki izin yang memadai untuk membuat server baru.

Solusi: Buka AWS OpsWorks for Chef Automate konsol; gunakan konsol untuk menghasilkan peran layanan baru dan peran profil instance. Jika Anda lebih suka menggunakan peran layanan Anda sendiri, lampirkan kebijakan CMServicePeran AWSOps Kerja ke peran tersebut. Verifikasi bahwa opsworks-cm.amazonaws.com terdaftar di antara layanan dalam hubungan Trust peran. Verifikasi bahwa peran layanan yang terkait dengan server Chef memiliki kebijakan terkelola CMServicePeran AWSOps Kerja yang dilampirkan.

Batas alamat IP elastis terlampaui

Masalah: Pembuatan server gagal dengan pesan kesalahan yang menyatakan, “Sumber daya berikut gagal dibuat: [EIP, EC2 Instance]. Pembuatan sumber daya dibatalkan, jumlah maksimum alamat telah tercapai.”

Penyebab: Ini terjadi ketika akun Anda telah menggunakan jumlah maksimum alamat Elastic IP (EIP). Batas alamat EIP default adalah lima.

Solusi: Anda dapat merilis alamat EIP yang ada atau menghapus alamat yang tidak digunakan secara aktif oleh akun Anda, atau Anda dapat menghubungi AWS Customer Support untuk meningkatkan batas alamat EIP yang terkait dengan akun Anda.

Tidak dapat masuk ke dasbor Chef Automate

Masalah: Dasbor Chef Automate menunjukkan kesalahan yang mirip dengan berikut ini: “Permintaan Lintas Asal Diblokir: Kebijakan Asal yang Sama melarang membaca sumber daya jarak jauh pada 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 binasi yang dimasukkan tidak benar.”

Penyebab: Eksplisit dasbor Chef Automate menetapkan FQDN, dan tidak menerima relatif. URLs Saat ini, Anda tidak dapat masuk dengan menggunakan alamat IP server Chef; Anda hanya dapat masuk dengan menggunakan nama DNS server.

Solusi: Masuk ke dasbor Chef Automate hanya dengan menggunakan entri nama DNS server Chef, bukan alamat IP-nya. Anda juga dapat mencoba mengatur ulang kredensyal dasbor Chef Automate dengan menjalankan AWS CLI perintah, seperti yang dijelaskan dalam. Atur Ulang Kredensi Dasbor Otomatis Chef

Asosiasi simpul tanpa pengawasan gagal

Masalah: Asosiasi EC2 node Amazon baru yang tidak dijaga, atau otomatis, gagal. Node yang seharusnya ditambahkan ke server Chef tidak muncul di dasbor Chef Automate, dan tidak tercantum dalam hasil `knife node show` perintah `knife client show` atau.

Penyebab: Hal ini dapat terjadi ketika Anda tidak memiliki peran IAM yang disiapkan sebagai profil instance yang memungkinkan panggilan `opsworks-cm` API untuk berkomunikasi dengan instance baru EC2 .

Solusi: Lampirkan kebijakan ke profil EC2 instans Anda yang memungkinkan panggilan `DescribeNodeAssociationStatus` API `AssociateNode` dan berfungsi EC2, seperti yang dijelaskan dalamTambahkan node secara otomatis di AWS OpsWorks for Chef Automate.

Pemeliharaan sistem gagal

AWS OpsWorks CM melakukan pemeliharaan sistem mingguan untuk memastikan bahwa versi minor terbaru dari Chef Server dan Chef Automate Server, termasuk pembaruan keamanan, selalu berjalan di server AWS OpsWorks untuk Chef Automate. Jika, karena alasan apa pun, pemeliharaan sistem gagal, AWS OpsWorks CM memberi tahu Anda tentang kegagalan tersebut. Untuk informasi selengkapnya tentang pemeliharaan sistem, lihatPemeliharaan Sistem di AWS OpsWorks for Chef Automate.

Bagian ini menjelaskan kemungkinan alasan kegagalan dan menyarankan solusi.

Topik

- Peran layanan atau kesalahan profil instance mencegah pemeliharaan sistem

Peran layanan atau kesalahan profil instance mencegah pemeliharaan sistem

Masalah: Pemeliharaan sistem gagal dengan pesan kesalahan yang menyatakan, “Tidak diizinkan untuk melakukan sts: AssumeRole “, atau pesan kesalahan serupa tentang izin.

Penyebab: Ini dapat terjadi ketika peran layanan atau profil instance yang Anda gunakan tidak memiliki izin yang memadai untuk melakukan pemeliharaan sistem di server.

Solusi: Pastikan bahwa peran layanan dan profil instans Anda memiliki semua izin yang diperlukan.

1. Pastikan bahwa peran layanan Anda memiliki semua izin yang diperlukan.

    a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk peran layanan yang digunakan server. Ini membuka peran layanan untuk dilihat di konsol IAM.

    b. Pada tab Izin, verifikasi yang `AWSOpsWorksCMServiceRole` dilampirkan ke peran layanan. Jika `AWSOpsWorksCMServiceRole` tidak terdaftar, tambahkan kebijakan ini ke peran.

    c. Verifikasi bahwa opsworks-cm.amazonaws.com terdaftar di antara layanan dalam hubungan Trust peran. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

2. Pastikan bahwa profil instans Anda memiliki semua izin yang diperlukan.

    a. Pada halaman Pengaturan untuk server Anda, di Jaringan dan keamanan, pilih tautan untuk profil instance yang digunakan server. Ini membuka profil instance untuk dilihat di konsol IAM.

    b. Pada tab Izin, verifikasi itu `AmazonEC2RoleforSSM` dan keduanya `AWSOpsWorksCMInstanceProfileRole` ada di daftar Kebijakan Izin. Jika salah satu atau keduanya tidak terdaftar, tambahkan kebijakan terkelola ini secara manual ke peran.

    c. Pada tab Hubungan kepercayaan, verifikasi bahwa peran layanan memiliki kebijakan kepercayaan yang mempercayai `ec2.amazonaws.com` layanan untuk mengambil peran atas nama Anda. Untuk informasi selengkapnya tentang cara menggunakan kebijakan kepercayaan dengan peran, lihat Memodifikasi peran (konsol), atau posting Blog AWS Keamanan, Cara menggunakan kebijakan kepercayaan dengan peran IAM.

Bantuan dan dukungan tambahan

Jika Anda tidak melihat masalah spesifik Anda yang dijelaskan dalam topik ini, atau Anda telah mencoba saran dalam topik ini dan masih mengalami masalah, kunjungi AWS OpsWorks forum.

Anda juga dapat mengunjungi AWS Support Center. AWS Support Center adalah hub untuk membuat dan mengelola kasus AWS Support. AWS Support Center juga menyertakan tautan ke sumber daya bermanfaat lainnya, seperti forum, teknis FAQs, status kesehatan layanan, dan AWS Trusted Advisor.

Keamanan dalam Manajemen AWS OpsWorks Konfigurasi (CM)

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara berkala menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program kepatuhan AWS. Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS OpsWorks CM, lihat AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS OpsWorks CM. Topik berikut menunjukkan cara mengonfigurasi AWS OpsWorks CM untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan layanan AWS lain yang membantu Anda memantau dan mengamankan sumber daya AWS OpsWorks CM Anda.

Topik

- Perlindungan Data di AWS OpsWorks CM
- Enkripsi data
- Identity and Access Management untuk AWS OpsWorks CM
- Privasi Lalu Lintas Kerja Internet
- Penebangan dan Pemantauan di AWS OpsWorks CM
- Validasi Kepatuhan untuk CM AWS OpsWorks
- Ketahanan dalam CM AWS OpsWorks

- Keamanan Infrastruktur di AWS OpsWorks CM
- Analisis Konfigurasi dan Kerentanan di CM AWS OpsWorks
- Praktik Terbaik Keamanan untuk AWS OpsWorks CM

Perlindungan Data di AWS OpsWorks CM

Model tanggung jawab AWS bersama model tanggung berlaku untuk perlindungan data di AWS OpsWorks Configuration Management. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan OpsWorks CM atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Nama-nama server OpsWorks CM tidak dienkripsi.

OpsWorks CM mengumpulkan data pelanggan berikut dalam proses membuat dan memelihara Anda AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise server.

- OpsWorks Untuk Puppet Enterprise, kami mengumpulkan kunci pribadi yang digunakan Puppet Enterprise untuk memungkinkan komunikasi antara master Wayang Anda dan node terkelola.
- Untuk AWS OpsWorks for Chef Automate, kami mengumpulkan kunci pribadi untuk sertifikat yang Anda lampirkan ke layanan jika Anda menggunakan domain khusus. Kunci pribadi yang Anda berikan saat membuat server Chef Automate dengan domain khusus diteruskan ke server Anda.

OpsWorks Server CM menyimpan kode konfigurasi Anda, seperti buku masak Chef atau modul Puppet Enterprise. Meskipun kode ini disimpan dalam backup server, AWS tidak memiliki akses ke sana. Konten ini dienkripsi, dan hanya administrator di AWS akun Anda yang dapat mengaksesnya. Kami menyarankan Anda mengamankan kode konfigurasi Chef atau Puppet Anda menggunakan protokol yang direkomendasikan untuk repositori sumber Anda. Misalnya, Anda dapat membatasi izin ke repositori AWS CodeCommit, atau mengikuti panduan di GitHub situs web untuk mengamankan repositori. GitHub

OpsWorks CM tidak menggunakan konten yang disediakan pelanggan untuk memelihara layanan, atau menyimpan catatan pelanggan. Log tentang server OpsWorks CM Anda disimpan di akun Anda, di bucket Amazon S3. Alamat IP pengguna yang terhubung ke server OpsWorks CM Anda dicatat oleh AWS.

Integrasi dengan AWS Secrets Manager

Mulai 3 Mei 2021, saat Anda membuat server baru di OpsWorks CM, OpsWorks CM menyimpan rahasia untuk server di AWS Secrets Manager. Untuk server baru, atribut berikut disimpan sebagai rahasia di Secrets Manager.

- Chef Mengotomatiskan server
  - Kunci pribadi HTTPS (hanya server yang tidak menggunakan domain khusus)
  - Chef Mengotomatiskan kata sandi administratif (CHEF_AUTOMATE_ADMIN_PASSWORD)
- Master Perusahaan Boneka
  - Kunci pribadi HTTPS (hanya server yang tidak menggunakan domain khusus)
  - Kata sandi administratif boneka (PUPPET_ADMIN_PASSWORD)
  - Remote boneka r10k (PUPPET_R10K_REMOTE)

Untuk server yang ada yang tidak menggunakan domain khusus, satu-satunya rahasia yang disimpan di Secrets Manager, untuk server Chef Automate dan Puppet Enterprise, adalah kunci pribadi HTTPS, karena ini dihasilkan selama pemeliharaan sistem mingguan otomatis.

OpsWorks CM menyimpan rahasia di Secrets Manager secara otomatis, dan perilaku ini tidak dapat dikonfigurasi pengguna.

Enkripsi data

AWS OpsWorks CM mengenkripsi cadangan server dan komunikasi antara AWS pengguna yang berwenang dan server CM mereka. AWS OpsWorks Namun, volume root Amazon EBS server AWS OpsWorks CM tidak dienkripsi.

Enkripsi saat Data Tidak Berpindah

AWS OpsWorks Cadangan server CM dienkripsi. Namun, volume root Amazon EBS server AWS OpsWorks CM tidak dienkripsi. Ini tidak dapat dikonfigurasi pengguna.

Enkripsi Saat Data Berpindah

AWS OpsWorks CM menggunakan HTTP dengan enkripsi TLS. AWS OpsWorks CM default ke sertifikat yang ditandatangani sendiri untuk menyediakan dan mengelola server, jika tidak ada sertifikat yang ditandatangani yang disediakan oleh pengguna. Kami menyarankan Anda menggunakan sertifikat yang ditandatangani oleh otoritas sertifikat (CA).

Manajemen kunci

AWS Key Management Service kunci terkelola pelanggan dan kunci terkelola AWS saat ini tidak didukung oleh AWS OpsWorks CM.

Identity and Access Management untuk AWS OpsWorks CM

AWS Identity and Access Management (IAM) adalah AWS layanan yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya CM. OpsWorks IAM adalah AWS layanan yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi Menggunakan Identitas
- Mengelola Akses Menggunakan Kebijakan
- Bagaimana AWS OpsWorks CM Bekerja dengan IAM
- AWS OpsWorks Contoh Kebijakan Berbasis Identitas CM
- Pemecahan Masalah AWS OpsWorks CM Identitas dan Akses
- AWS kebijakan terkelola untuk Manajemen AWS OpsWorks Konfigurasi
- Cross-service membingungkan deputi pencegahan di AWS OpsWorks CM

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di OpsWorks CM.

Pengguna layanan — Jika Anda menggunakan layanan OpsWorks CM untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur OpsWorks CM untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di OpsWorks CM, lihatPemecahan Masalah AWS OpsWorks CM Identitas dan Akses.

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya OpsWorks CM di perusahaan Anda, Anda mungkin memiliki akses penuh ke OpsWorks CM. Tugas Anda adalah menentukan fitur dan sumber daya OpsWorks CM mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan OpsWorks CM, lihatBagaimana AWS OpsWorks CM Bekerja dengan IAM.

Administrator IAM - Jika Anda seorang administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke OpsWorks CM. Untuk melihat contoh kebijakan berbasis identitas OpsWorks CM yang dapat Anda gunakan di IAM, lihat. AWS OpsWorks Contoh Kebijakan Berbasis Identitas CM

Mengautentikasi Menggunakan Identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun

AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Pengguna dan Grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

> **Warning**
>
> Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan

hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

- Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola Akses Menggunakan Kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat

kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan Berbasis Identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat dilampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

OpsWorks CM mendukung kebijakan khusus yang Anda buat di IAM dan dilampirkan ke pengguna, peran, atau grup.

Kebijakan Berbasis Sumber Daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

OpsWorks CM tidak mendukung kebijakan berbasis sumber daya.

Daftar Kontrol Akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

OpsWorks CM tidak digunakan ACLs.

Tipe Kebijakan Lainnya

OpsWorks CM tidak mendukung jenis kebijakan lain berikut ini.

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batas izin — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran). Anda dapat menetapkan batas izin untuk suatu entitas. Izin yang dihasilkan adalah persimpangan antara kebijakan berbasis identitas milik entitas dan batas izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola beberapa AWS akun secara terpusat yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Cara SCPs Kerja di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi.

Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai Tipe Kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana AWS OpsWorks CM Bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke AWS OpsWorks CM, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan dengan AWS OpsWorks CM. Untuk mendapatkan tampilan tingkat tinggi tentang bagaimana AWS OpsWorks CM dan AWS layanan lainnya bekerja dengan IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Topik

- AWS OpsWorks Kebijakan Berbasis Identitas CM
- AWS OpsWorks CM dan Kebijakan Berbasis Sumber Daya
- Otorisasi Berdasarkan Tag AWS OpsWorks CM
- AWS OpsWorks Peran CM IAM

AWS OpsWorks Kebijakan Berbasis Identitas CM

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. AWS OpsWorks CM mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan IAM JSON dalam Panduan Pengguna IAM.

Di AWS OpsWorks CM, Anda dapat melampirkan pernyataan kebijakan khusus ke pengguna, peran, atau grup.

Tindakan

Elemen `Action` kebijakan berbasis identitas IAM menjelaskan tindakan atau tindakan tertentu yang akan diizinkan atau ditolak oleh kebijakan tersebut. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Tindakan ini digunakan dalam kebijakan untuk memberikan izin guna melakukan operasi terkait.

Tindakan kebijakan di AWS OpsWorks CM menggunakan awalan berikut sebelum tindakan:`opsworks-cm:`. Misalnya, untuk memberikan izin kepada seseorang untuk membuat server AWS OpsWorks CM menggunakan operasi API, Anda menyertakan `opsworks-cm:CreateServer` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus mencakup salah satu `Action` atau `NotAction` elemen. AWS OpsWorks CM mendefinisikan serangkaian tindakannya sendiri yang menggambarkan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
      "opsworks-cm:action1",
      "opsworks-cm:action2"
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "opsworks-cm:Describe*"
```

Saat Anda menggunakan wildcard untuk mengizinkan beberapa tindakan dalam pernyataan kebijakan, berhati-hatilah agar Anda mengizinkan tindakan tersebut hanya untuk layanan atau pengguna resmi.

Untuk melihat daftar tindakan AWS OpsWorks CM, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk AWS OpsWorks di Panduan Pengguna IAM.

Sumber daya

Elemen `Resource` menentukan objek di mana tindakan berlaku. Pernyataan harus mencakup elemen `Resource` atau `NotResource`. Anda menentukan sumber daya menggunakan ARN atau menggunakan wildcard (*) untuk menunjukkan bahwa pernyataan berlaku untuk semua sumber daya.

Anda bisa mendapatkan Amazon Resource Number (ARN) dari server AWS OpsWorks CM atau cadangan dengan menjalankan operasi DescribeServersatau DescribeBackupsAPI, dan mendasarkan kebijakan tingkat sumber daya pada sumber daya tersebut.

Sumber daya server AWS OpsWorks CM memiliki ARN dalam format berikut:

```
arn:aws:opsworks-cm:{Region}:${Account}:server/${ServerName}/${UniqueId}
```

Sumber daya cadangan AWS OpsWorks CM memiliki ARN dalam format berikut:

```
arn:aws:opsworks-cm:{Region}:${Account}:backup/${ServerName}-{Date-and-Time-Stamp-of-Backup}
```

Untuk informasi selengkapnya tentang format ARNs, lihat Amazon Resource Names (ARNs) dan Ruang Nama AWS Layanan.

Misalnya, untuk menentukan server `test-chef-automate` Chef Automate dalam pernyataan Anda, gunakan ARN berikut:

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/test-chef-automate/EXAMPLE-d1a2bEXAMPLE"
```

Untuk menentukan semua server AWS OpsWorks CM milik akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:server/*"
```

Contoh berikut menentukan cadangan server AWS OpsWorks CM sebagai sumber daya:

```
"Resource": "arn:aws:opsworks-cm:us-west-2:123456789012:backup/test-chef-automate-server-2018-05-20T19:06:12.399Z"
```

Beberapa tindakan AWS OpsWorks CM, seperti untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

```
"Resource": "*"
```

Banyak tindakan API melibatkan beberapa sumber daya. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARNs dengan koma.

```
"Resource": [
      "resource1",
      "resource2"
```

Untuk melihat daftar jenis sumber daya AWS OpsWorks CM dan jenisnya ARNs, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk AWS OpsWorks CM di Panduan Pengguna IAM. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk AWS OpsWorks CM di Panduan Pengguna IAM.

Kunci kondisi

AWS OpsWorks CM tidak memiliki kunci konteks khusus layanan yang dapat digunakan dalam `Condition` elemen pernyataan kebijakan. Untuk daftar kunci konteks global yang tersedia untuk semua layanan, lihat kunci konteks kondisi AWS global di Referensi Kebijakan IAM. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, seperti sama dengan atau kurang dari, untuk mencocokkan ketentuan dalam kebijakan dengan nilai dalam permintaan.

Jika Anda menentukan beberapa elemen `Condition` dalam pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS mengevaluasinya menggunakan operasi `AND`. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama pengguna. Untuk informasi lebih lanjut, lihat Elemen kebijakan IAM: Variabel dan tanda dalam Panduan Pengguna IAM.

Contoh

Untuk melihat contoh kebijakan berbasis identitas AWS OpsWorks CM, lihat. AWS OpsWorks Contoh Kebijakan Berbasis Identitas CM

AWS OpsWorks CM dan Kebijakan Berbasis Sumber Daya

AWS OpsWorks CM tidak mendukung kebijakan berbasis sumber daya.

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya dan dalam kondisi apa.

Otorisasi Berdasarkan Tag AWS OpsWorks CM

Anda dapat melampirkan tag ke sumber daya AWS OpsWorks CM atau meneruskan tag dalam permintaan ke AWS OpsWorks CM. Untuk mengontrol akses berdasarkan tag, Anda memberikan informasi tag dalam elemen kondisi kebijakan menggunakan`aws:RequestTag/key-name`, atau kunci `aws:TagKeys` kondisi. Untuk informasi selengkapnya tentang menandai sumber daya AWS OpsWorks CM, lihat Bekerja dengan Tag pada AWS OpsWorks for Chef Automate Sumber Daya atau Bekerja dengan Tag pada AWS OpsWorks for Puppet Enterprise Sumber Daya di panduan ini.

AWS OpsWorks Peran CM IAM

Peran IAM adalah entitas dalam AWS akun Anda yang memiliki izin tertentu.

AWS OpsWorks CM menggunakan dua peran:

- Peran layanan yang memberikan izin layanan AWS OpsWorks CM untuk bekerja dalam akun pengguna. AWS Jika Anda menggunakan peran layanan default yang disediakan oleh OpsWorks CM, nama peran ini adalah`aws-opsworks-cm-service-role`.
- Peran profil instance yang memungkinkan layanan AWS OpsWorks CM memanggil OpsWorks CM API. Peran ini memberikan akses ke Amazon S3 AWS CloudFormation dan untuk membuat server dan bucket S3 untuk cadangan. Jika Anda menggunakan profil instans default yang disediakan oleh OpsWorks CM, nama peran profil instance ini adalah`aws-opsworks-cm-ec2-role`.

AWS OpsWorks CM tidak menggunakan peran terkait layanan.

Menggunakan Kredensial Sementara dengan CM AWS OpsWorks

AWS OpsWorks CM mendukung penggunaan kredensi sementara, dan mewarisi kemampuan itu dari. AWS Security Token Service

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensil keamanan sementara dengan memanggil operasi AWS STS API seperti AssumeRoleatau. GetFederationToken

Peran Tertaut Layanan

AWS OpsWorks CM tidak menggunakan peran terkait layanan.

Peran terkait AWS layanan memungkinkan layanan mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran terkait layanan muncul di akun IAM Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Peran Layanan

Fitur ini memungkinkan layanan untuk menerima peran layanan atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di akun IAM Anda dan dimiliki oleh akun tersebut. Ini berarti administrator IAM dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

AWS OpsWorks CM menggunakan dua peran:

- Peran layanan yang memberikan izin layanan AWS OpsWorks CM untuk bekerja dalam akun pengguna. AWS Jika Anda menggunakan peran layanan default yang disediakan oleh OpsWorks CM, nama peran ini adalah`aws-opsworks-cm-service-role`.
- Peran profil instance yang memungkinkan layanan AWS OpsWorks CM memanggil OpsWorks CM API. Peran ini memberikan akses ke Amazon S3 AWS CloudFormation dan untuk membuat server dan bucket S3 untuk cadangan. Jika Anda menggunakan profil instans default yang disediakan oleh OpsWorks CM, nama peran profil instance ini adalah`aws-opsworks-cm-ec2-role`.

Memilih Peran IAM dalam CM AWS OpsWorks

Saat membuat server di AWS OpsWorks CM, Anda harus memilih peran untuk mengizinkan AWS OpsWorks CM mengakses Amazon EC2 atas nama Anda. Jika Anda telah membuat peran layanan, maka AWS OpsWorks CM memberi Anda daftar peran untuk dipilih. OpsWorks CM dapat membuat peran untuk Anda, jika Anda tidak menentukan satu. Penting untuk memilih peran yang memungkinkan akses untuk memulai dan menghentikan EC2 instans Amazon. Untuk informasi selengkapnya, lihat Buat Server Otomatis Koki atau Buat Master Perusahaan Boneka.

AWS OpsWorks Contoh Kebijakan Berbasis Identitas CM

Secara default, pengguna atau peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS OpsWorks CM. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS CLI, atau AWS API. Administrator IAM harus membuat kebijakan IAM yang memberikan izin identitas IAM untuk melakukan operasi API tertentu pada sumber daya tertentu

yang mereka butuhkan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM di Panduan Pengguna IAM.

Di AWS OpsWorks CM, Anda dapat menetapkan `AWSOpsWorksCMServiceRole` kebijakan kepada pengguna agar pengguna dapat membuat dan mengelola server Chef Automate atau Puppet Enterprise menggunakan server atau. AWS Management Console AWS CLI

Topik

- Praktik Terbaik Kebijakan
- Izinkan Pengguna untuk Melihat Izin Mereka Sendiri
- Melihat Server AWS OpsWorks CM Berdasarkan Tag

Praktik Terbaik Kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya OpsWorks CM di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber

daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.

- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Izinkan Pengguna untuk Melihat Izin Mereka Sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "ViewOwnUserInfo",
               "Effect": "Allow",
               "Action": [
                   "iam:GetUserPolicy",
                   "iam:ListGroupsForUser",
                   "iam:ListAttachedUserPolicies",
```

```
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Melihat Server AWS OpsWorks CM Berdasarkan Tag

Anda dapat menggunakan kondisi dalam kebijakan berbasis identitas untuk mengontrol akses ke server AWS OpsWorks CM dan backup berdasarkan tag. Contoh ini menunjukkan cara membuat kebijakan yang memungkinkan melihat server AWS OpsWorks CM. Namun, izin diberikan hanya jika tag server AWS OpsWorks CM `Owner` memiliki nilai nama pengguna pengguna tersebut. Kebijakan ini juga memberi izin yang diperlukan untuk menyelesaikan tindakan ini pada konsol tersebut.

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

Anda dapat melampirkan kebijakan ini ke pengguna di akun Anda. Jika pengguna bernama `richard-roe` mencoba untuk melihat server AWS OpsWorks CM, server harus diberi tag `Owner=richard-roe` atau`owner=richard-roe`. Jika tidak, aksesnya akan ditolak. Kunci tanda syarat `Owner` cocok dengan `Owner` dan `owner` karena nama kunci syarat tidak terpengaruh huruf besar/kecil. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM JSON: Syarat dalam Panduan Pengguna IAM.

Pemecahan Masalah AWS OpsWorks CM Identitas dan Akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan IAM. Untuk informasi pemecahan masalah khusus untuk AWS OpsWorks CM, lihat Pemecahan masalah AWS OpsWorks for Chef Automate dan. Pemecahan Masalah OpsWorks untuk Perusahaan Boneka

Topik

- Saya Tidak Berwenang untuk Melakukan Tindakan di AWS OpsWorks CM
- Saya Tidak Berwenang untuk Melakukan iam: PassRole
- Saya Ingin Mengizinkan Orang Di Luar AWS Akun Saya Mengakses Sumber Daya AWS OpsWorks CM Saya

Saya Tidak Berwenang untuk Melakukan Tindakan di AWS OpsWorks CM

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Contoh kesalahan berikut terjadi ketika pengguna `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang server AWS OpsWorks CM, tetapi tidak memiliki `opsworks-cm:DescribeServers` izin.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 opsworks-cm:DescribeServers on resource: test-chef-automate-server
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakan untuk memungkinkannya mengakses `test-chef-automate-server` sumber daya menggunakan `opsworks-cm:DescribeServers` tindakan tersebut.

Saya Tidak Berwenang untuk Melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diberi otorisasi untuk melakukan tindakan `iam:PassRole`, Anda harus menghubungi administrator untuk mendapatkan bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk. Minta orang tersebut untuk memperbarui kebijakan Anda agar Anda dapat meneruskan peran ke OpsWorks CM.

Beberapa AWS layanan memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut, alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di OpsWorks CM. Namun, tindakan ini mengharuskan layanan memiliki izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut ke layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, Mary meminta administrator untuk memperbarui kebijakannya agar dia dapat melakukan tindakan `iam:PassRole`.

Saya Ingin Mengizinkan Orang Di Luar AWS Akun Saya Mengakses Sumber Daya AWS OpsWorks CM Saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis

sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- AWS OpsWorks CM mendukung pemberian pengguna dari lebih dari satu akses akun untuk mengelola server AWS OpsWorks CM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh AWS akun yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di AWS akun lain yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda ke AWS akun pihak ketiga, lihat Menyediakan akses ke AWS akun yang dimiliki oleh pihak ketiga di Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.

AWS kebijakan terkelola untuk Manajemen AWS OpsWorks Konfigurasi

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan yang dikelola pelanggan IAM yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di AWS akun Anda. Untuk informasi selengkapnya tentang kebijakan AWS AWS terkelola, lihat kebijakan terkelola di Panduan Pengguna IAM.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan yang dikelola AWS untuk mendukung fitur-fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan yang dikelola AWS saat ada fitur baru yang diluncurkan atau saat ada operasi baru yang tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan ReadOnlyAccess AWS terkelola menyediakan akses hanya-baca ke

semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk melihat daftar dan deskripsi dari kebijakan fungsi tugas, lihat kebijakan yang dikelola AWS untuk fungsi tugas di Panduan Pengguna IAM.

Kebijakan terkelola AWS: **`AWSOpsWorksCMServiceRole`**

Anda dapat melampirkan `AWSOpsWorksCMServiceRole` ke entitas IAM Anda. OpsWorks CM juga melampirkan kebijakan ini ke peran layanan yang memungkinkan OpsWorks CM melakukan tindakan atas nama Anda.

Kebijakan ini memberikan *`administrative`* izin yang memungkinkan administrator OpsWorks CM membuat, mengelola, dan menghapus server dan cadangan OpsWorks CM.

Detail izin

Kebijakan ini mencakup izin berikut.

- `opsworks-cm`— Memungkinkan prinsipal untuk menghapus server yang ada, dan memulai pemeliharaan berjalan.
- `acm`— Memungkinkan prinsipal untuk menghapus atau mengimpor sertifikat dari AWS Certificate Manager yang memungkinkan pengguna terhubung ke server CM. OpsWorks
- `cloudformation`— Memungkinkan OpsWorks CM untuk membuat dan mengelola AWS CloudFormation tumpukan ketika prinsipal membuat, memperbarui, atau menghapus server CM. OpsWorks
- `ec2`— Memungkinkan OpsWorks CM meluncurkan, menyediakan, memperbarui, dan menghentikan instans Amazon Elastic Compute Cloud saat prinsipal membuat, memperbarui, atau menghapus server CM. OpsWorks
- `iam`— Memungkinkan OpsWorks CM untuk membuat peran layanan yang diperlukan untuk membuat dan mengelola server OpsWorks CM.
- `tag`— Memungkinkan prinsipal untuk menerapkan dan menghapus tag dari sumber daya OpsWorks CM, termasuk server dan cadangan.
- `s3`— Memungkinkan OpsWorks CM membuat bucket Amazon S3 untuk menyimpan cadangan server, mengelola objek di bucket S3 berdasarkan permintaan utama (misalnya, menghapus cadangan), dan menghapus bucket.
- `secretsmanager`— Memungkinkan OpsWorks CM untuk membuat dan mengelola rahasia Secrets Manager, dan menerapkan atau menghapus tag dari rahasia.

- `ssm`— Memungkinkan OpsWorks CM untuk menggunakan Systems Manager Run Command pada instance yang merupakan server OpsWorks CM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket*"
            ],
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteObject",
                "s3:DeleteBucket",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutBucketPolicy",
                "s3:PutObject",
                "s3:GetBucketTagging",
                "s3:PutBucketTagging"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "tag:UntagResources",
                "tag:TagResources"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:GetCommandInvocation",
                "ssm:ListCommandInvocations",
```

```
                "ssm:ListCommands"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:cloudformation:stack-name": "aws-opsworks-cm-
*"
                }
            },
            "Action": [
                "ssm:SendCommand"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*::document/*",
                "arn:aws:s3:::amzn-s3-demo-bucket*"
            ],
            "Action": [
                "ssm:SendCommand"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Action": [
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateImage",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSnapshot",
                "ec2:CreateTags",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteSnapshot",
                "ec2:DeregisterImage",
```

```
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSnapshots",
                "ec2:DescribeSubnets",
                "ec2:DisassociateAddress",
                "ec2:ReleaseAddress",
                "ec2:RunInstances",
                "ec2:StopInstances"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ec2:ResourceTag/aws:cloudformation:stack-name": "aws-opsworks-cm-
*"
                }
            },
            "Action": [
                "ec2:TerminateInstances",
                "ec2:RebootInstances"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:opsworks-cm:*:*:server/*"
            ],
            "Action": [
                "opsworks-cm:DeleteServer",
                "opsworks-cm:StartMaintenance"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/aws-opsworks-cm-*"
```

```
            ],
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/aws-opsworks-cm-*",
                "arn:aws:iam::*:role/service-role/aws-opsworks-cm-*"
            ],
            "Action": [
                "iam:PassRole"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "acm:DeleteCertificate",
                "acm:ImportCertificate"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "arn:aws:secretsmanager:*:*:opsworks-cm!aws-opsworks-cm-
secrets-*",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:TagResource",
                "secretsmanager:UntagResource"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteTags",
```

```
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:elastic-ip/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

Kebijakan terkelola AWS: **AWSOpsWorksCMInstanceProfileRole**

Anda dapat melampirkan `AWSOpsWorksCMInstanceProfileRole` ke entitas IAM Anda. OpsWorks CM juga melampirkan kebijakan ini ke peran layanan yang memungkinkan OpsWorks CM melakukan tindakan atas nama Anda.

Kebijakan ini memberikan *`administrative`* izin yang memungkinkan EC2 instans Amazon yang digunakan sebagai server OpsWorks CM untuk mendapatkan informasi dari AWS CloudFormation dan AWS Secrets Manager menyimpan cadangan server di bucket Amazon S3.

Detail izin

Kebijakan ini mencakup izin berikut.

- `acm`— Memungkinkan EC2 instance server OpsWorks CM untuk mendapatkan sertifikat dari AWS Certificate Manager yang memungkinkan pengguna terhubung ke server OpsWorks CM.
- `cloudformation`— Memungkinkan EC2 instance server OpsWorks CM untuk mendapatkan informasi tentang AWS CloudFormation tumpukan selama proses pembuatan atau pembaruan instans, dan mengirim sinyal ke AWS CloudFormation tentang statusnya.
- `s3`— Memungkinkan EC2 instance server OpsWorks CM untuk mengunggah dan menyimpan cadangan server di bucket S3, menghentikan atau memutar kembali unggahan jika perlu, dan menghapus cadangan dari bucket S3.
- `secretsmanager`— Memungkinkan EC2 instance server OpsWorks CM untuk mendapatkan nilai rahasia Secrets Manager terkait 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 CM memperbarui kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk OpsWorks CM sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen OpsWorks CM.

Perubahan	Deskripsi	Tanggal
AWSOpsBekerja CMInstance ProfileRole - Kebijakan terkelola yang diperbarui	OpsWorks CM memperbarui kebijakan terkelola yang memungkinkan EC2 instance yang digunakan sebagai server OpsWorks CM untuk berbagi informasi dengan CloudFormation Secrets Manager, dan mengelola cadangan. Perubahan menambah `opsworks-cm!` nama sumber daya untuk rahasia Secrets Manager, sehingga OpsWorks CM diizinkan untuk memiliki rahasia.	23 April 2021
AWSOpsCMServicePeranKerja - Kebijakan terkelola yang diperbarui	OpsWorks CM memperbarui kebijakan terkelola yang memungkinkan administrator OpsWorks CM membuat, mengelola, dan menghapus server dan cadangan OpsWorks CM. Perubahan menambah `opsworks-cm!` nama sumber daya untuk rahasia Secrets Manager, sehingga OpsWorks CM diizinkan untuk memiliki rahasia.	23 April 2021
OpsWorks CM mulai melacak perubahan	OpsWorks CM mulai melacak perubahan untuk kebijakan AWS terkelolanya.	23 April 2021

Cross-service membingungkan deputi pencegahan di AWS OpsWorks CM

Masalah "confused deputy" adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang memilik hak akses lebih tinggi untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan dalam kebijakan sumber daya untuk membatasi izin yang AWS OpsWorks CM memberikan layanan lain ke sumber daya. Jika nilai `aws:SourceArn` tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global tersebut untuk membatasi izin. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama. Gunakan `aws:SourceArn` jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Nilai `aws:SourceArn` harus ARN dari OpsWorks CM Chef atau Puppet server.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci konteks kondisi `aws:SourceArn` global dengan ARN penuh server. AWS OpsWorks CM Jika Anda tidak mengetahui ARN lengkap, atau jika Anda menentukan beberapa server ARNs, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (`*`) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Bagian berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan AWS OpsWorks CM untuk mencegah masalah wakil yang membingungkan.

Mencegah eksploitasi wakil yang bingung di AWS OpsWorks CM

Bagian ini menjelaskan bagaimana Anda dapat membantu mencegah eksploitasi deputi yang membingungkan AWS OpsWorks CM, dan menyertakan contoh kebijakan izin yang dapat Anda

lampirkan ke peran IAM yang Anda gunakan untuk mengakses. AWS OpsWorks CM Sebagai praktik terbaik keamanan, kami sarankan untuk menambahkan kunci `aws:SourceArn` dan `aws:SourceAccount` kondisi ke hubungan kepercayaan yang dimiliki peran IAM Anda dengan layanan lain. Hubungan kepercayaan memungkinkan AWS OpsWorks CM untuk mengambil peran untuk melakukan tindakan di layanan lain yang diperlukan untuk membuat atau mengelola AWS OpsWorks CM server Anda.

Untuk mengedit hubungan kepercayaan untuk menambahkan **`aws:SourceArn`** dan **`aws:SourceAccount`** mengkondisikan kunci

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Peran.
3. Di kotak Pencarian, cari peran yang Anda gunakan untuk akses AWS OpsWorks CM. Peran yang AWS dikelola adalah`aws-opsworks-cm-service-role`.
4. Pada halaman Ringkasan untuk peran tersebut, pilih tab Trust relationship.
5. Pilih tab Hubungan Kepercayaan, pilih Edit Hubungan Kepercayaan.
6. Dalam dokumen Kebijakan, tambahkan setidaknya satu dari `aws:SourceArn` atau kunci `aws:SourceAccount` kondisi ke kebijakan. Gunakan `aws:SourceArn` untuk membatasi hubungan kepercayaan antara layanan lintas (seperti AWS Certificate Manager dan Amazon EC2) dan AWS OpsWorks CM ke AWS OpsWorks CM server tertentu, yang lebih ketat. Tambahkan `aws:SourceAccount` untuk membatasi hubungan kepercayaan antara lintas layanan dan AWS OpsWorks CM server di akun tertentu, yang kurang membatasi. Berikut adalah contohnya. Perhatikan bahwa jika Anda menggunakan kedua tombol kondisi, akun IDs harus sama.

```
{
  "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. Setelah selesai menambahkan kunci kondisi, pilih Perbarui kebijakan kepercayaan.

Berikut ini adalah contoh tambahan peran yang membatasi akses ke AWS OpsWorks CM server dengan menggunakan `aws:SourceArn` dan`aws:SourceAccount`.

Topik

- Contoh: Mengakses AWS OpsWorks CM server di wilayah tertentu
- Contoh: Menambahkan lebih dari satu ARN server ke aws:SourceArn

Contoh: Mengakses AWS OpsWorks CM server di wilayah tertentu

Pernyataan hubungan kepercayaan peran berikut mengakses AWS OpsWorks CM server mana pun di Wilayah AS Timur (Ohio) (`us-east-2`). Perhatikan bahwa wilayah ditentukan dalam nilai ARN`aws:SourceArn`, tetapi nilai ID server adalah wildcard (*).

```
{
  "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/*"
        }
      }
```

```
    }
  ]
}
```

Contoh: Menambahkan lebih dari satu ARN server ke **`aws:SourceArn`**

Contoh berikut membatasi akses ke array dua AWS OpsWorks CM server di ID akun 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"
           ]
       }
      }
    }
  ]
}
```

Privasi Lalu Lintas Kerja Internet

AWS OpsWorks CM menggunakan protokol keamanan transmisi yang sama yang umumnya digunakan oleh AWS: HTTPS, atau HTTP dengan enkripsi TLS.

Penebangan dan Pemantauan di AWS OpsWorks CM

AWS OpsWorks CM mencatat semua tindakan API ke CloudTrail. Untuk informasi selengkapnya, lihat topik berikut.

- Logging OpsWorks untuk Panggilan API Perusahaan Boneka dengan AWS CloudTrail
- Pencatatan Panggilan AWS OpsWorks for Chef Automate API dengan AWS CloudTrail

Validasi Kepatuhan untuk CM AWS OpsWorks

AWS OpsWorks CM mendukung program dan peraturan kepatuhan berikut:

- Industri Kartu Pembayaran (Payment Card Industry/PCI)
- Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan 1996 (HIPAA)
- AWS Sistem dan Kontrol Organisasi (SOC) 1, 2, dan 3
- Peraturan Perlindungan Data Umum (GDPR)

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS OpsWorks CM sebagai bagian dari beberapa program AWS kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat AWS Services in Scope by Compliance Program. Untuk informasi umum, lihat Program Kepatuhan AWS.

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda saat menggunakan AWS OpsWorks CM ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan memberikan langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan di AWS.
- Arsitektur untuk Whitepaper Keamanan dan Kepatuhan HIPAA — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS

- AWS Sumber Daya Kepatuhan — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config AWS Layanan ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Ketahanan dalam CM AWS OpsWorks

AWS OpsWorks CM memungkinkan backup harian server secara default ketika Anda membuat server. Cadangan dienkripsi dan disimpan dalam bucket Amazon S3. Secara default, bucket ini hanya dapat diakses oleh akun yang membuat server. Anda dapat menambahkan akses bucket untuk pengguna lain atau mengonfigurasi cadangan lintas wilayah di Amazon S3 sesuai kebijaksanaan Anda. Chef dan Puppet mendukung enkripsi lintas wilayah, karena kedua produk mengenkripsi lalu lintas antara server AWS OpsWorks CM Anda dan node terkelola.

AWS OpsWorks CM tidak mendukung konfigurasi ketersediaan tinggi (HA).

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Zona Ketersediaan tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur biasa yang terdiri dari satu atau beberapa pusat data.

Untuk informasi selengkapnya tentang cara mencadangkan dan memulihkan server di AWS OpsWorks CM, lihat berikut ini:

- Cadangkan dan Kembalikan Server OpsWorks untuk Puppet Enterprise
- Membuat cadangan dan Memulihkan AWS OpsWorks for Chef Automate Server

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan Infrastruktur di AWS OpsWorks CM

Sebagai layanan terkelola, AWS OpsWorks Configuration Management dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses OpsWorks CM melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

AWS OpsWorks CM tidak mendukung tautan pribadi atau titik akhir pribadi VPC.

AWS OpsWorks CM tidak mendukung kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat AWS Layanan yang Bekerja dengan IAM di Panduan AWS Identity and Access Management Pengguna.

Analisis Konfigurasi dan Kerentanan di CM AWS OpsWorks

AWS OpsWorks CM melakukan pembaruan kernel dan keamanan berkala ke sistem operasi yang berjalan di server AWS OpsWorks CM Anda. Pengguna dapat mengatur jendela waktu agar pembaruan otomatis terjadi hingga dua minggu dari tanggal saat ini. AWS OpsWorks CM mendorong pembaruan otomatis versi minor Chef dan Puppet Enterprise. Untuk informasi selengkapnya tentang mengonfigurasi pembaruan AWS OpsWorks for Chef Automate, lihat Pemeliharaan Sistem (Chef) dalam panduan ini. Untuk informasi selengkapnya tentang mengonfigurasi pembaruan OpsWorks untuk Puppet Enterprise, lihat Pemeliharaan Sistem (Boneka) dalam panduan ini.

Praktik Terbaik Keamanan untuk AWS OpsWorks CM

AWS OpsWorks CM, seperti semua AWS layanan, menawarkan fitur keamanan untuk dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai pertimbangan yang bermanfaat, bukan sebagai resep.

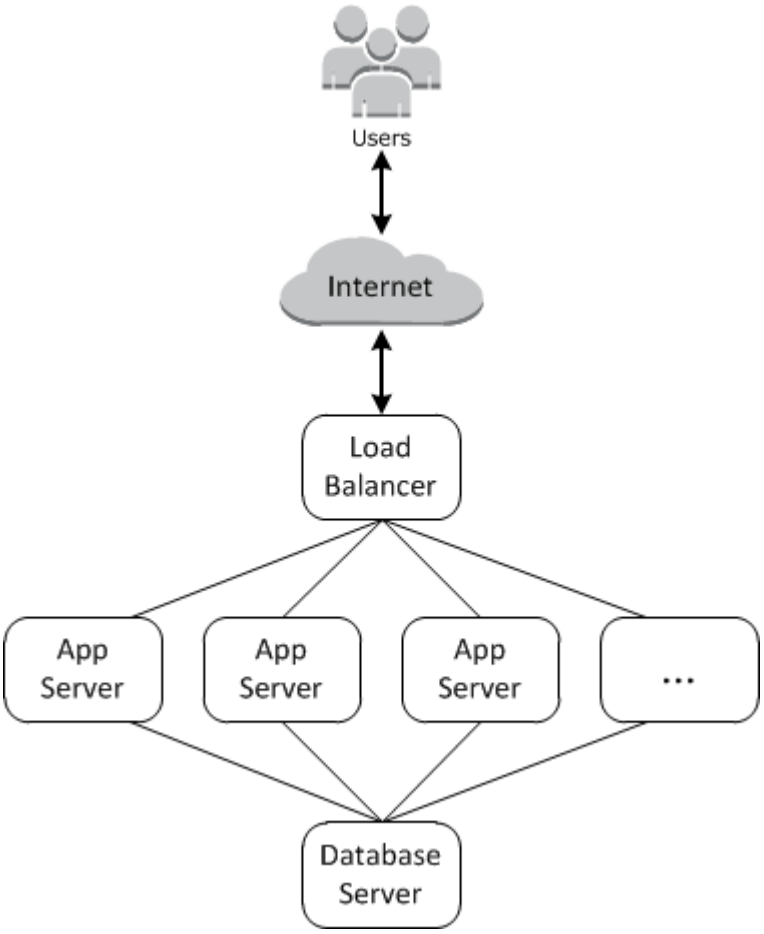
- Amankan Starter Kit Anda dan unduh kredensyal login. Saat Anda membuat server AWS OpsWorks CM baru atau mengunduh Starter Kit dan kredensyal baru dari konsol AWS OpsWorks CM, simpan item ini di lokasi aman yang memerlukan setidaknya satu faktor otentikasi minimal. Kredensialnya menyediakan akses tingkat administrator ke server Anda.
- Amankan kode konfigurasi Anda. Amankan kode konfigurasi Chef atau Puppet Anda (buku masak dan modul) menggunakan protokol yang direkomendasikan untuk repositori sumber Anda. Misalnya, Anda dapat membatasi izin ke repositori AWS CodeCommit, atau mengikuti panduan di GitHub situs web untuk mengamankan repositori. GitHub
- Gunakan sertifikat yang ditandatangani CA untuk terhubung ke node. Meskipun Anda dapat menggunakan sertifikat yang ditandatangani sendiri saat Anda mendaftar atau mem-boot node di server AWS OpsWorks CM Anda, sebagai praktik terbaik, gunakan sertifikat yang ditandatangani CA. Kami menyarankan Anda menggunakan sertifikat yang ditandatangani oleh otoritas sertifikat (CA).
- Jangan bagikan kredensyal masuk konsol manajemen Chef atau Puppet dengan pengguna lain. Administrator harus membuat pengguna terpisah untuk setiap pengguna situs web konsol Chef atau Puppet.
  - Kelola Pengguna di Chef Automate
  - Mengelola Pengguna di Perusahaan Boneka
- Konfigurasikan backup otomatis dan pembaruan pemeliharaan sistem. Mengkonfigurasi pembaruan pemeliharaan otomatis pada server AWS OpsWorks CM Anda membantu memastikan bahwa server Anda menjalankan pembaruan sistem operasi terkait keamanan terbaru. Mengkonfigurasi backup otomatis membantu memudahkan pemulihan bencana dan mempercepat waktu pemulihan jika terjadi insiden atau kegagalan. Batasi akses ke bucket Amazon S3 yang menyimpan cadangan server AWS OpsWorks CM Anda; jangan berikan akses ke Semua Orang. Berikan akses baca atau tulis ke pengguna lain secara individual sesuai kebutuhan, atau buat grup keamanan di IAM untuk pengguna tersebut, dan tetapkan akses ke grup keamanan.
  - Pemeliharaan Sistem (Chef)

- Pemeliharaan Sistem (Boneka)
- Membuat cadangan dan Memulihkan AWS OpsWorks for Chef Automate Server
- Cadangkan dan Kembalikan Server OpsWorks untuk Puppet Enterprise
- Membuat Pengguna dan Grup Delegasi IAM Pertama Anda di Panduan Pengguna AWS Identity and Access Management
- Praktik Terbaik Keamanan untuk Amazon S3 di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon

AWS OpsWorks Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

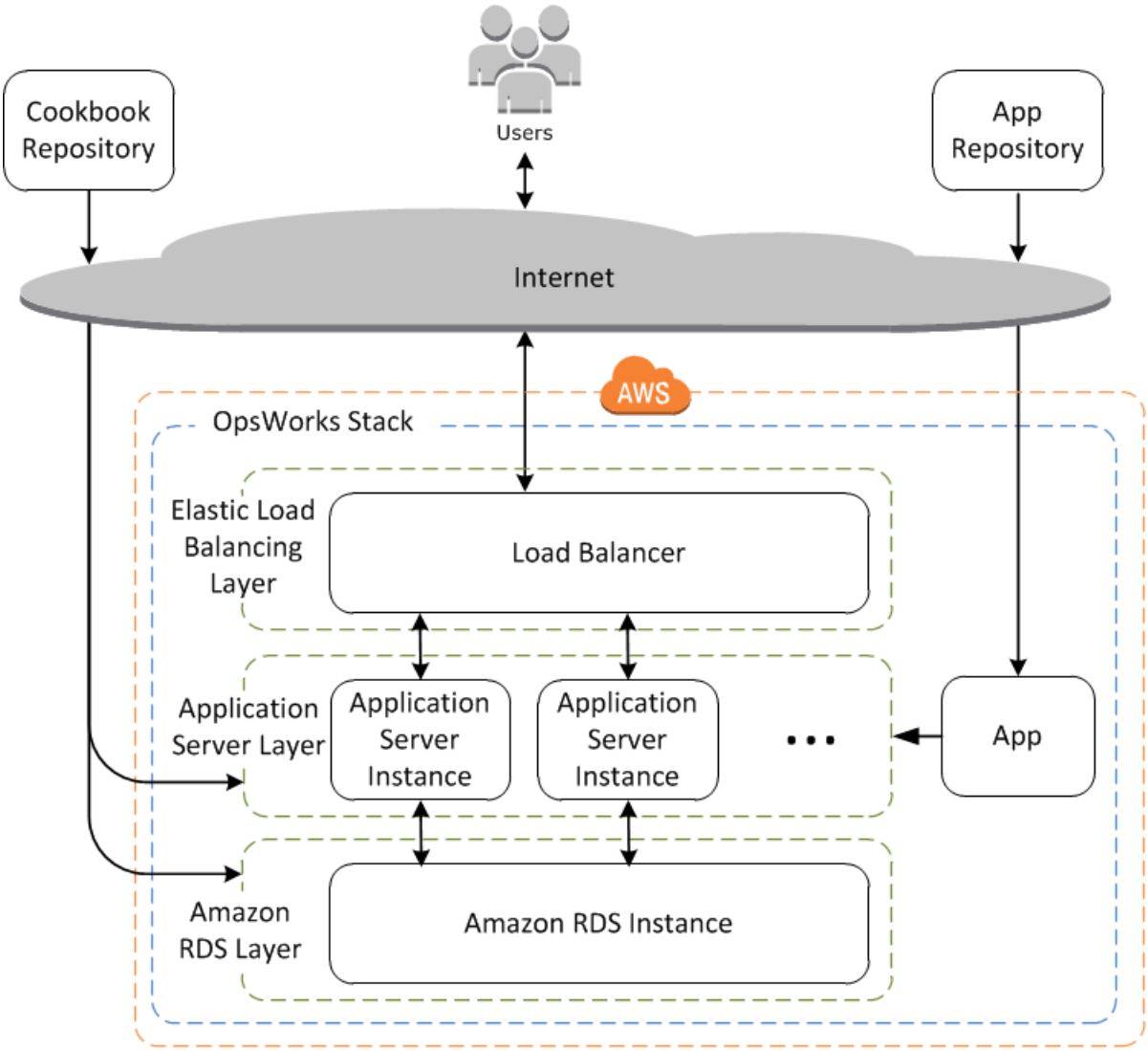
Komputasi berbasis cloud biasanya melibatkan grup sumber daya AWS, seperti instans Amazon dan EC2 instans Amazon Relational Database Service (RDS), yang harus dibuat dan dikelola secara kolektif. Misalnya, aplikasi web biasanya membutuhkan server aplikasi, server database, penyeimbang beban, dan sebagainya. Kelompok contoh ini biasanya disebut tumpukan; tumpukan server aplikasi sederhana mungkin terlihat seperti berikut ini.



Selain membuat instance dan menginstal paket yang diperlukan, Anda biasanya memerlukan cara untuk mendistribusikan aplikasi ke server aplikasi, memantau kinerja tumpukan, mengelola keamanan dan izin, dan sebagainya.

AWS OpsWorks Stacks menyediakan cara sederhana dan fleksibel untuk membuat dan mengelola tumpukan dan aplikasi.

Berikut adalah bagaimana tumpukan server aplikasi dasar mungkin terlihat dengan AWS OpsWorks Stacks. Ini terdiri dari sekelompok server aplikasi yang berjalan di belakang penyeimbang beban Elastic Load Balancing, dengan server database Amazon RDS backend.



Meskipun relatif sederhana, tumpukan ini menunjukkan semua fitur AWS OpsWorks Stacks utama. Begini caranya disatukan.

Topik

- Tumpukan
- Lapisan
- Resep dan LifeCycle Acara
- Instans
- Aplikasi
- Menyesuaikan tumpukan Anda
- Manajemen Sumber Daya
- Keamanan dan Izin
- Pemantauan dan Pencatatan
- CLI, SDK, dan Template AWS CloudFormation
- AWS OpsWorks Stacks Akhir Kehidupan FAQs
- Migrasi AWS OpsWorks Stacks aplikasi Anda ke Manajer AWS Systems Manager Aplikasi
- Menggunakan alat AWS OpsWorks Stacks Detach in Place
- Memulai dengan AWS OpsWorks Stacks
- AWS OpsWorks Praktik Terbaik Stacks
- Tumpukan
- Lapisan
- Instans
- Aplikasi
- Buku Masak dan Resep
- Manajemen Sumber Daya
- Tanda
- Pemantauan
- Keamanan dan Izin
- AWS OpsWorks Dukungan Stacks untuk Chef 12 Linux
- Support untuk Versi Chef Sebelumnya di AWS OpsWorks Stacks
- Menggunakan AWS OpsWorks Stacks dengan Layanan AWS Lainnya
- Menggunakan AWS OpsWorks Stacks CLI
- Panduan Debugging dan Pemecahan Masalah
- AWS OpsWorks Agen Tumpukan CLI

- AWS OpsWorks Referensi Tas Data Stacks
- OpsWorks Perubahan Agen

Tumpukan

Tumpukan adalah komponen AWS OpsWorks Stacks inti. Ini pada dasarnya adalah wadah untuk AWS Resources—instans Amazon, EC2 instans database Amazon RDS, dan sebagainya—yang memiliki tujuan bersama dan harus dikelola secara logis bersama. Tumpukan membantu Anda mengelola sumber daya ini sebagai grup dan juga menentukan beberapa pengaturan konfigurasi default, seperti sistem operasi instans dan wilayah AWS. Jika Anda ingin mengisolasi beberapa komponen tumpukan dari interaksi pengguna langsung, Anda dapat menjalankan tumpukan di VPC.

Lapisan

Anda menentukan konstituen tumpukan dengan menambahkan satu atau lebih lapisan. Lapisan mewakili satu set EC2 instance Amazon yang melayani tujuan tertentu, seperti melayani aplikasi atau hosting server database.

Anda dapat menyesuaikan atau memperluas lapisan dengan memodifikasi konfigurasi default paket, menambahkan resep Chef untuk melakukan tugas-tugas seperti menginstal paket tambahan, dan banyak lagi.

Untuk semua tumpukan, AWS OpsWorks Stacks menyertakan lapisan layanan, yang mewakili layanan AWS berikut.

- Amazon Relational Database Service
- Penyeimbang Beban Elastis
- Amazon Elastic Container Service

Lapisan memberi Anda kontrol penuh atas paket mana yang diinstal, bagaimana mereka dikonfigurasi, bagaimana aplikasi digunakan, dan banyak lagi.

Resep dan LifeCycle Acara

Lapisan bergantung pada resep Chef untuk menangani tugas seperti menginstal paket pada instance, menerapkan aplikasi, menjalankan skrip, dan sebagainya. Salah satu fitur utama AWS

OpsWorks Stacks adalah serangkaian peristiwa siklus hidup —Setup, Configure, Deploy, Undeploy, dan Shutdown—yang secara otomatis menjalankan serangkaian resep tertentu pada waktu yang tepat pada setiap instance.

Setiap lapisan dapat memiliki satu set resep yang ditetapkan untuk setiap peristiwa siklus hidup, yang menangani berbagai tugas untuk acara dan lapisan itu. Misalnya, setelah instance milik lapisan server web selesai booting, AWS OpsWorks Stacks melakukan hal berikut.

1. Menjalankan resep Setup layer, yang dapat melakukan tugas-tugas seperti menginstal dan mengkonfigurasi server web.
2. Menjalankan resep Deploy layer, yang menyebarkan aplikasi layer dari repositori ke instance dan melakukan tugas terkait, seperti memulai ulang layanan.
3. Menjalankan resep Konfigurasi pada setiap instance di tumpukan sehingga setiap instance dapat menyesuaikan konfigurasinya sesuai kebutuhan untuk mengakomodasi instance baru.

   Misalnya, pada instance yang menjalankan penyeimbang beban, resep Configure dapat memodifikasi konfigurasi load balancer untuk menyertakan instance baru.

Jika sebuah instance milik beberapa lapisan, AWS OpsWorks Stacks menjalankan resep untuk setiap lapisan sehingga Anda dapat, misalnya, memiliki instance yang mendukung server aplikasi PHP dan server database MySQL.

Jika Anda telah menerapkan resep, Anda dapat menetapkan setiap resep ke lapisan dan acara yang sesuai dan AWS OpsWorks Stacks secara otomatis menjalankannya untuk Anda pada waktu yang tepat. Anda juga dapat menjalankan resep secara manual, kapan saja.

Instans

Sebuah instance mewakili sumber daya komputasi tunggal, seperti EC2 instans Amazon. Ini mendefinisikan konfigurasi dasar sumber daya, seperti sistem operasi dan ukuran. Pengaturan konfigurasi lainnya, seperti alamat IP Elastis atau volume Amazon EBS, ditentukan oleh lapisan instans. Resep lapisan menyelesaikan konfigurasi dengan melakukan tugas-tugas seperti menginstal dan mengonfigurasi paket dan menerapkan aplikasi.

Anda dapat menggunakan AWS OpsWorks Stacks untuk membuat instance dan menambahkannya ke layer. Saat memulai instance, AWS OpsWorks Stacks meluncurkan EC2 instance Amazon menggunakan pengaturan konfigurasi yang ditentukan oleh instance dan lapisannya. Setelah

EC2 instans Amazon selesai booting, AWS OpsWorks Stacks menginstal agen yang menangani komunikasi antara instance dan layanan dan menjalankan resep yang sesuai sebagai respons terhadap peristiwa siklus hidup.

AWS OpsWorks Stacks mendukung jenis contoh berikut, yang ditandai dengan bagaimana mereka dimulai dan dihentikan.

- Instance 24/7 dimulai secara manual dan dijalankan sampai Anda menghentikannya.
- Instans berbasis waktu dijalankan oleh AWS OpsWorks Stacks pada jadwal harian dan mingguan yang ditentukan.

  Mereka memungkinkan tumpukan Anda untuk secara otomatis menyesuaikan jumlah instance untuk mengakomodasi pola penggunaan yang dapat diprediksi.
- Instans berbasis beban secara otomatis dimulai dan dihentikan oleh AWS OpsWorks Stacks, berdasarkan metrik beban yang ditentukan, seperti pemanfaatan CPU.

  Mereka memungkinkan tumpukan Anda untuk secara otomatis menyesuaikan jumlah instance untuk mengakomodasi variasi lalu lintas masuk. Instans berbasis beban hanya tersedia untuk tumpukan berbasis Linux.

AWS OpsWorks Stacks mendukung penyembuhan otomatis instance. Jika agen berhenti berkomunikasi dengan layanan, AWS OpsWorks Stacks secara otomatis berhenti dan memulai ulang instance.

Anda juga dapat menggabungkan sumber daya komputasi berbasis Linux ke dalam tumpukan yang dibuat di luar Stacks. AWS OpsWorks

- EC2 Instans Amazon yang Anda buat langsung dengan menggunakan EC2 konsol Amazon, CLI, atau API.
- Instans lokal yang berjalan di perangkat keras Anda sendiri, termasuk instance yang berjalan di mesin virtual.

Setelah Anda mendaftarkan salah satu contoh ini, itu menjadi instance AWS OpsWorks Stacks dan Anda dapat mengelolanya dengan cara yang sama seperti contoh yang Anda buat dengan Stacks. AWS OpsWorks

Aplikasi

Anda menyimpan aplikasi dan file terkait di repositori, seperti bucket Amazon S3. Setiap aplikasi diwakili oleh aplikasi, yang menentukan jenis aplikasi dan berisi informasi yang diperlukan untuk menyebarkan aplikasi dari repositori ke instance Anda, seperti URL repositori dan kata sandi. Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks akan memicu peristiwa Deploy, yang menjalankan resep Deploy pada instance tumpukan.

Anda dapat menerapkan aplikasi dengan cara berikut:

- Secara otomatis—Saat memulai instance, AWS OpsWorks Stacks secara otomatis menjalankan resep Deploy instans.
- Secara manual—Jika Anda memiliki aplikasi baru atau ingin memperbarui aplikasi yang sudah ada, Anda dapat menjalankan resep Deploy instans online secara manual.

Anda biasanya memiliki AWS OpsWorks Stacks menjalankan resep Deploy di seluruh tumpukan, yang memungkinkan instance lapisan lain untuk memodifikasi konfigurasi mereka dengan tepat. Namun, Anda dapat membatasi penerapan ke subset instance jika, misalnya, Anda ingin menguji aplikasi baru sebelum menerapkannya ke setiap instance server aplikasi.

Menyesuaikan tumpukan Anda

AWS OpsWorks Stacks menyediakan berbagai cara untuk menyesuaikan lapisan untuk memenuhi kebutuhan spesifik Anda:

- Anda dapat mengubah cara AWS OpsWorks Stacks mengonfigurasi paket dengan mengganti atribut yang mewakili berbagai pengaturan konfigurasi, atau bahkan dengan mengganti templat yang digunakan untuk membuat file konfigurasi.
- Anda dapat memperluas lapisan yang ada dengan menyediakan resep Anda sendiri untuk melakukan tugas-tugas seperti menjalankan skrip atau menginstal dan mengkonfigurasi paket yang tidak standar.

Semua tumpukan dapat mencakup satu atau lebih lapisan, yang dimulai dengan hanya satu set resep minimal. Anda menambahkan fungsionalitas ke lapisan dengan menerapkan resep untuk menangani tugas-tugas seperti menginstal paket, menerapkan aplikasi, dan sebagainya. Anda mengemas resep khusus dan file terkait Anda dalam satu atau beberapa buku masak dan menyimpan buku masak di repositori seperti Amazon S3 atau Git.

Anda dapat menjalankan resep secara manual, tetapi AWS OpsWorks Stacks juga memungkinkan Anda mengotomatiskan proses dengan mendukung serangkaian lima peristiwa siklus hidup:

- Pengaturan terjadi pada instance baru setelah berhasil melakukan boot.
- Konfigurasi terjadi pada semua instance tumpukan saat instance masuk atau keluar dari status online.
- Deploy terjadi saat Anda menerapkan aplikasi.
- Undeploy terjadi saat Anda menghapus aplikasi.
- Shutdown terjadi ketika Anda menghentikan sebuah instance.

Setiap lapisan dapat memiliki berapa pun jumlah resep yang ditetapkan untuk setiap acara. Ketika peristiwa siklus hidup terjadi pada instance layer, AWS OpsWorks Stacks menjalankan resep terkait. Misalnya, saat peristiwa Deploy terjadi pada instance server aplikasi, AWS OpsWorks Stacks menjalankan resep Deploy layer untuk mengunduh aplikasi atau melakukan tugas terkait.

Manajemen Sumber Daya

Anda dapat memasukkan sumber daya AWS lainnya, seperti alamat IP Elastic, ke dalam tumpukan Anda. Anda dapat menggunakan konsol AWS OpsWorks Stacks atau API untuk mendaftarkan sumber daya dengan tumpukan, melampirkan sumber daya terdaftar ke atau melepaskannya dari instance, dan memindahkan sumber daya dari satu instance ke instance lainnya.

Keamanan dan Izin

AWS OpsWorks Stacks terintegrasi dengan AWS Identity and Access Management (IAM) untuk menyediakan cara yang kuat untuk mengontrol cara pengguna mengakses AWS OpsWorks Stacks, termasuk yang berikut:

- Bagaimana pengguna individu dapat berinteraksi dengan setiap tumpukan, seperti apakah mereka dapat membuat sumber daya tumpukan seperti lapisan dan instance, atau apakah mereka dapat menggunakan SSH atau RDP untuk terhubung ke instance Amazon tumpukan. EC2
- Bagaimana AWS OpsWorks Stacks dapat bertindak atas nama Anda untuk berinteraksi dengan sumber daya AWS seperti EC2 instans Amazon.
- Bagaimana aplikasi yang berjalan pada instans AWS OpsWorks Stacks dapat mengakses sumber daya AWS seperti bucket Amazon S3.

- Cara mengelola kunci SSH publik pengguna dan kata sandi RDP dan terhubung ke sebuah instance.

Pemantauan dan Pencatatan

AWS OpsWorks Stacks menyediakan beberapa fitur untuk membantu Anda memantau tumpukan Anda dan memecahkan masalah dengan tumpukan Anda dan resep apa pun. Untuk semua tumpukan:

- AWS OpsWorks Stacks menyediakan satu set CloudWatch metrik kustom untuk tumpukan Linux, yang dirangkum untuk kenyamanan Anda di halaman Monitoring.

  AWS OpsWorks Stacks mendukung CloudWatch metrik standar untuk tumpukan Windows. Anda dapat memantaunya dengan CloudWatch konsol.
- CloudTrail log, yang merekam panggilan API yang dibuat oleh atau atas nama AWS OpsWorks Tumpukan di akun AWS Anda.
- Log peristiwa, yang mencantumkan semua peristiwa di tumpukan Anda.
- Chef mencatat detail apa yang terjadi untuk setiap peristiwa siklus hidup pada setiap instance, seperti resep mana yang dijalankan dan kesalahan mana yang terjadi.

Tumpukan berbasis Linux juga dapat menyertakan lapisan master Ganglia, yang dapat Anda gunakan untuk mengumpulkan dan menampilkan data pemantauan terperinci untuk instance di tumpukan Anda.

CLI, SDK, dan Template AWS CloudFormation

Selain konsol, AWS OpsWorks Stacks juga mendukung antarmuka baris perintah (CLI) dan SDKs untuk beberapa bahasa yang dapat digunakan untuk melakukan operasi apa pun. Pertimbangkan fitur-fitur ini:

- AWS OpsWorks Stacks CLI adalah bagian dari AWS CLI, dan dapat digunakan untuk melakukan operasi apa pun dari baris perintah.

  AWS CLI mendukung beberapa layanan AWS dan dapat diinstal pada sistem Windows, Linux, atau OS X.

- AWS OpsWorks Stacks disertakan dalam AWS Tools untuk Windows PowerShell dan dapat digunakan untuk melakukan operasi apa pun dari baris PowerShell perintah Windows.
- AWS OpsWorks Stacks SDK disertakan dalam AWS SDKs, yang dapat digunakan oleh aplikasi yang diimplementasikan di: Java, JavaScript(berbasis browser dan Node.js), .NET, PHP, Python (boto), atau Ruby.

Anda juga dapat menggunakan AWS CloudFormation templat untuk menyediakan tumpukan. Untuk beberapa contoh, lihat AWS OpsWorks Snippets.

AWS OpsWorks Stacks Akhir Kehidupan FAQs

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin.

Topik

- Bagaimana pelanggan yang sudah ada akan terpengaruh oleh End of Life ini?
- Apakah AWS OpsWorks Stacks menerima pelanggan baru?
- Ke mana saya harus memigrasikan tumpukan yang ada?
- Bagaimana saya bisa menyimpan EC2 instance Amazon saya yang ada setelah End of Life?
- Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama?
- Tingkat dukungan teknis apa yang tersedia AWS OpsWorks Stacks?
- Apakah akan ada rilis fitur baru untuk AWS OpsWorks Stacks?

Bagaimana pelanggan yang sudah ada akan terpengaruh oleh End of Life ini?

Pelanggan yang ada tidak akan terpengaruh hingga 26 Mei 2024, tanggal Akhir Masa Pakai untuk. AWS OpsWorks Stacks Setelah 26 Mei 2024, pelanggan tidak akan dapat menggunakan OpsWorks konsol, API, CLI, CloudFormation dan sumber daya.

Apakah AWS OpsWorks Stacks menerima pelanggan baru?

Tidak. AWS OpsWorks Stacks tidak lagi menerima pelanggan baru dan hanya pelanggan yang sudah ada yang dapat membuat tumpukan baru saat ini.

Ke mana saya harus memigrasikan tumpukan yang ada?

Kami menyarankan AWS OpsWorks Stacks pelanggan memigrasikan beban kerja mereka ke AWS Systems Manager tempat mereka dapat memanfaatkan fitur-fitur berikut:

- Versi Chef modern
- SSM Agent
- Application Load Balancer
- Fitur penskalaan yang disempurnakan melalui grup Auto Scaling
- Kemampuan untuk menentukan karakteristik host yang diinginkan menggunakan template EC2 peluncuran
- Jenis contoh yang lebih baru
- Jenis volume EBS yang lebih baru

Untuk informasi tentang Systems Manager, lihat Panduan AWS Systems Manager Pengguna. Untuk informasi tentang migrasi ke AWS Systems Manager, lihat Migrasi AWS OpsWorks Stacks aplikasi Anda ke Manajer AWS Systems Manager Aplikasi

Bagaimana saya bisa menyimpan EC2 instance Amazon saya yang ada setelah End of Life?

Setelah tanggal Akhir Masa Pakai tercapai, EC2 instans Amazon Anda akan tetap ada di akun Anda, tetapi Anda tidak lagi dapat menggunakan layanan OpsWorks Stacks untuk mengontrol dan mengelola instans.

Anda dapat menggunakan alat AWS OpsWorks Stacks Lepaskan di Tempat untuk melepaskan OpsWorks instance Anda dari layanan Stacks. OpsWorks Setelah detasemen, Anda dapat menggunakan Amazon EC2, AWS Systems Manager, atau pendekatan apa pun yang EC2 kompatibel untuk mengonfigurasi dan mengelola instance. Untuk informasi selengkapnya, lihat Menggunakan alat AWS OpsWorks Stacks Detach in Place.

Akankah Akhir Kehidupan mempengaruhi semua Wilayah AWS pada saat yang sama?

Ya. OpsWorks Konsol, API, CLI, dan CloudFormation sumber daya akan dihentikan Wilayah AWS secara bersamaan pada 26 Mei 2024. Untuk daftar Wilayah AWS tempat AWS OpsWorks Stacks tersedia, lihat Daftar Layanan AWS Regional.

Tingkat dukungan teknis apa yang tersedia AWS OpsWorks Stacks?

AWS akan terus memberikan tingkat dukungan yang sama untuk AWS OpsWorks Stacks yang dimiliki pelanggan hari ini hingga tanggal Akhir Kehidupan. Jika Anda memiliki pertanyaan atau masalah, Anda dapat menghubungi AWS Dukungan Tim di AWS RE:Post atau melalui AWS Dukungan Premium.

Apakah akan ada rilis fitur baru untuk AWS OpsWorks Stacks?

Tidak. Karena layanan ini mencapai End of Life, kami tidak akan merilis fitur baru apa pun. Namun, kami akan terus melakukan peningkatan keamanan dan mengelola EC2 instans Amazon seperti yang diharapkan hingga tanggal Akhir Kehidupan.

Migrasi AWS OpsWorks Stacks aplikasi Anda ke Manajer AWS Systems Manager Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin.

Anda sekarang dapat memigrasikan AWS OpsWorks Stacks aplikasi Anda ke Manajer Aplikasi, kemampuan AWS Systems Manager, menggunakan skrip migrasi. Memigrasikan aplikasi Stacks ke Manajer Aplikasi Systems Manager memungkinkan Anda menggunakan AWS fitur yang tidak tersedia AWS OpsWorks Stacks, seperti jenis EC2 instans Amazon baru seperti Graviton, volume Amazon Elastic Block Store (EBS) baru seperti gp3, sistem operasi baru, integrasi dengan grup Auto Scaling, dan penyeimbang beban aplikasi.

Dengan rilis ini, Anda sekarang dapat memantau dan menjalankan operasi pada instance yang dimigrasi menggunakan tab Instances baru yang tersedia dari Systems Manager Application Manager. Anda dapat menggunakan tab Instances untuk melihat beberapa AWS instance di satu tempat. Dengan menggunakan tab ini, Anda dapat melihat informasi tentang kesehatan instans dan memecahkan masalah. Untuk informasi selengkapnya tentang bekerja dengan tab Instans, lihat Bekerja dengan instance aplikasi Anda di AWS Systems Manager Panduan Pengguna.

Topik

- Cara kerja skrip
- Prasyarat
- Batasan
- Memulai
- Pertanyaan yang Sering Diajukan
- Pemecahan Masalah

Cara kerja skrip

AWS OpsWorks menyediakan skrip yang dapat Anda jalankan untuk memigrasikan AWS OpsWorks Stacks aplikasi Anda ke Manajer Aplikasi Systems Manager menggunakan CloudFormation templat. Skrip mendapatkan informasi tentang OpsWorks lapisan yang ada dan tergantung pada nilai `--provision-application` parameter untuk skrip, baik menyediakan tiruan dari aplikasi Anda, atau menyediakan CloudFormation template pemula yang dapat Anda modifikasi menggunakan AWS CloudFormation.

Prasyarat

- Pastikan bahwa AWS CLI sudah diinstal dan dikonfigurasi. Untuk informasi selengkapnya tentang menginstal AWS CLI, lihat Menginstal atau memperbarui versi terbaru dari Panduan AWS Command Line Interface Pengguna. AWS CLI

> **Note**
>
> Jika Anda tidak ingin mengkonfigurasi AWS CLI, Anda juga dapat menjalankan perintah menggunakan AWS CloudShell. Untuk informasi selengkapnya tentang bekerja dengan CloudShell, lihat Bekerja dengan AWS CloudShell di Panduan AWS CloudShell Pengguna.

- Pastikan bahwa Python versi 3.6 atau yang lebih baru diinstal atau dilengkapi dengan Amazon Machine Image (AMI).
- Pastikan sistem operasi Anda didukung. Anda dapat mengunduh dan menjalankan skrip migrasi pada sistem operasi berikut.
  - Amazon Linux dan Amazon Linux 2
  - Ubuntu 18.04 LTS, 20.04 LTS, 22.04 LTS
  - Perusahaan Topi Merah Linux 8
  - Windows Server 2019, Windows 10 Perusahaan

> **Note**
>
> Windows Server 2022 tidak didukung.

Batasan

OpsWorks Arsitektur baru berbeda dari arsitektur untuk AWS OpsWorks Stacks. Bagian ini menjelaskan keterbatasan yang diketahui dari arsitektur ini.

Berikut ini tidak didukung oleh OpsWorks arsitektur baru.

- Menjalankan resep Chef di instance Windows dan CentOS
- Built-in Chef 11 lapisan dan Berkshelf
- Atribut koki dan kantong data
- Instans lokal
- Instans yang diimpor dari EC2
- Tidak ada dukungan untuk menginstal daftar paket sistem operasi yang ditentukan pengguna
- Aplikasi tidak didukung atau dimigrasikan

Berikut ini didukung dengan keterbatasan.

- Skrip migrasi mengkloning informasi volume EBS, tetapi tidak termasuk titik pemasangan dan data aktual yang terkandung dalam volume.

- Instance skala berbasis waktu dan berbasis beban dimigrasikan, tetapi aturan penskalaan apa pun yang terkait dengan instance ini tidak dimigrasikan. Anda dapat memodifikasi grup Auto Scaling untuk mencapai hasil yang serupa.
- Entitas IAM yang ditentukan di halaman Izin tumpukan di OpsWorks konsol tidak dibuat atau dibuat.
- Skrip migrasi hanya mampu menyediakan aplikasi lapisan tunggal di Systems Manager. Misalnya, jika Anda menjalankan skrip dua kali untuk dua lapisan dalam tumpukan yang sama, Anda mendapatkan dua aplikasi berbeda di Systems Manager.

Memulai

Skrip migrasi,`stack_exporter.py`, adalah skrip Python yang dapat Anda jalankan secara lokal, atau pada sebuah instance. EC2 Sebelum menjalankan skrip, pastikan semua prasyarat terpenuhi. Untuk lebih lanjut tentang prasyarat, lihat. Prasyarat

Langkah-langkah di bagian berikut menunjukkan cara memigrasikan OpsWorks tumpukan Anda ke Manajer Aplikasi Systems Manager.

Topik

- Langkah 1: Siapkan lingkungan Anda untuk menjalankan skrip
- Langkah 2: Unduh skrip migrasi
- Langkah 3: Siapkan lingkungan Anda untuk menjalankan skrip
- Langkah 4: Jalankan skrip
- Langkah 5: Menyediakan CloudFormation tumpukan
- Langkah 6: Tinjau sumber daya yang disediakan
- Langkah 7: Mulai sebuah instance
- Langkah 8: Tinjau instance
- Langkah 9: Memantau dan menjalankan operasi pada instans Anda menggunakan Systems Manager Application Manager

Langkah 1: Siapkan lingkungan Anda untuk menjalankan skrip

Persiapkan lingkungan Anda dengan menjalankan perintah yang sesuai untuk sistem operasi Anda.

Topik

- Amazon Linux 2
- Amazon Linux
- Ubuntu 18.04, 20.04, 22.04
- Perusahaan Topi Merah Linux 8
- Windows Server 2019, Windows 10 Perusahaan

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
```

Untuk Python versi 3.6, jalankan juga:

```
python3 -m pip install pipenv==2022.4.8
```

Untuk Python versi 3.7 dan yang lebih baru, jalankan juga:

```
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
```

Perusahaan Topi Merah 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 Perusahaan

> **Note**
>
> Untuk Windows Server 2019, instal Python versi 3.6.1 atau yang lebih baru.

```
pip install pipenv
```

Jika Git belum diinstal, unduh dan instal Git.

Jika Anda menggunakan Git sebagai sumber buku masak, tambahkan server Git Anda ke `known_hosts` file sebelum menjalankan skrip di Windows. Anda dapat menggunakan PowerShell untuk membuat fungsi berikut.

```
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."
   }
}
```

Anda kemudian dapat menyediakan server Git Anda (misalnya, github.com, git-codecommit.*repository_region*.amazonaws.com) ketika Anda menjalankan fungsi.

```
add_to_known_hosts "myGitServer"
```

Langkah 2: Unduh skrip migrasi

Unduh file zip yang berisi skrip migrasi dan semua file yang relevan dengan menjalankan perintah berikut.

```
aws s3api get-object \
    --bucket export-opsworks-stacks-bucket-prod-us-east-1 \
    --key export_opsworks_stacks_script.zip export_opsworks_stacks_script.zip
```

Jika Anda menggunakan Linux, instal utilitas unzip menggunakan perintah berikut.

```
sudo apt-get install unzip
sudo yum install unzip
```

Buka zip file menggunakan perintah yang sesuai untuk sistem operasi Anda.

Untuk Linux, gunakan perintah berikut.

```
unzip export_opsworks_stacks_script.zip
```

Untuk Windows, gunakan `Expand-Archive` perintah di PowerShell.

```
Expand-Archive -LiteralPath PathToZipFile -DestinationPath PathToDestination
```

Setelah file di-unzip, direktori dan file berikut tersedia.

- README.md
- LISENSI
- MELIHAT
- requirements.txt
- template/
  - OpsWorksCFNTemplate.yaml
  - Gunung EBSVolumes .yaml
- opsworks/
- cloudformation/
- instances_tab/
- cfn_stack_deployer.py
- s3.py
- stack_exporter_context.py
- stack_exporter.py

Langkah 3: Siapkan lingkungan Anda untuk menjalankan skrip

Siapkan lingkungan Anda untuk menjalankan skrip dengan menggunakan perintah berikut.

```
pipenv install -r requirements.txt
pipenv shell
```

> **Note**
>
> Saat ini, skrip hanya dapat menyediakan aplikasi single-layer di Application Manager. Misalnya, jika Anda menjalankan skrip dua kali untuk dua lapisan dalam tumpukan yang sama, skrip membuat dua aplikasi berbeda di Manajer Aplikasi.

Setelah mengatur lingkungan Anda, tinjau parameter skrip. Anda dapat melihat opsi yang tersedia untuk skrip migrasi dengan menjalankan `python3 stack_exporter.py --help` perintah.

Parameter	Deskripsi	Diperlukan	Tipe	Nilai default
`--layer-id`	Mengekspor CloudFormation template untuk ID OpsWorks layer ini.	Ya	string	
`--region`	AWS Wilayah untuk OpsWorks tumpukan. Jika OpsWorks Stack Region dan API endpoint Region berbeda, gunakan Stack Region. Ini adalah Wilayah yang sama dengan bagian sumber daya lain dari OpsWorks tumpukan Anda (misalnya, EC2 instance dan subnet).	Tidak	string	us-east-1
`--provision-application`	Secara default, skrip menyediakan aplikasi yang diekspor oleh CloudFormation template. Masukkan parameter ini ke dalam skrip dengan nilai FALSE untuk melewati penyediaan template. CloudFormation	Tidak	Boolean	`TRUE`
`--launch-template`	Parameter ini menentukan apakah akan menggunakan template peluncuran yang ada, atau membuat template peluncuran baru. Anda dapat membuat template peluncuran baru yang menggunakan properti instance yang direkomendasikan, atau yang menggunakan properti instance yang cocok dengan instance online. Nilai yang valid meliputi:	Tidak	string	`RECOMMENDED`

Parameter	Deskripsi	Diperlukan	Tipe	Nilai default
	• `RECOMMENDED` - Menggunakan karakteristik instance dari AMI terbaru untuk OS OpsWorks stack dan ukuran instans c5.large. • `MATCH_LAST_INSTANCE` - Menggunakan karakteristik instans online terbaru yang tersedia. • *`LaunchTemplateID`* `/` `[`*`LaunchTemplateVersion]`* - Menggunakan template peluncuran yang ada. Secara opsional, Anda dapat memberikan versi template. Jika Anda tidak menyediakan versi template, skrip menggunakan versi default.			

Parameter	Deskripsi	Diperlukan	Tipe	Nilai default
`--system-updates`	Mendefinisikan apakah akan melakukan pembaruan kernel dan paket saat instance boot. Nilai yang valid meliputi: • `ALL_UPDATES` - Melakukan pembaruan sistem untuk kernel dan paket saat instance boot. • `NO_UPDATES` - Tidak melakukan pembaruan sistem saat instance boot. • `MATCH_LAYER_SETTINGS` - Menggunakan `InstallUpdatesOnBoot` properti OpsWorks layer atau instance untuk menentukan apakah akan menginstal pembaruan sistem.	Tidak	string	`ALL_UPDATES`
`--http-username`	Nama `SecureString` parameter Systems Manager yang menyimpan nama pengguna yang digunakan untuk mengautentikasi ke arsip HTTP yang berisi buku masak kustom.	Tidak	string	
`--http-password`	Nama `SecureString` parameter Systems Manager yang menyimpan kata sandi yang digunakan untuk mengautentikasi ke arsip HTTP yang berisi buku masak kustom.	Tidak	string	

Parameter	Deskripsi	Diperlukan	Tipe	Nilai default
`--repo-private-key`	Nama `SecureString` parameter Systems Manager yang menyimpan kunci SSH yang digunakan untuk mengautentikasi ke repositori yang berisi buku masak kustom. Jika repositori aktif GitHub, Anda harus membuat kunci `Ed25519` SSH baru. Jika Anda tidak menghasilkan kunci `Ed25519` SSH baru, koneksi ke GitHub repositori gagal.	Tidak	string	
`--lb-type`	Jenis penyeimbang beban, jika ada, yang akan dibuat saat memigrasikan penyeimbang beban yang ada. Nilai yang valid meliputi: • `ALB`(Application Load Balancer) • `Classis`(Classic Load Balancer) • `None`(jika Anda tidak ingin membuat penyeimbang beban)	Tidak	string	`ALB`

Parameter	Deskripsi	Diperlukan	Tipe	Nilai default
`--lb-access-logs-path`	Jalur ke bucket dan awalan S3 yang ada untuk menyimpan log akses penyeimbang beban. Bucket S3 dan penyeimbang beban harus berada di Wilayah yang sama. Jika Anda tidak memberikan nilai dan nilai `--lb-type` parameter disetel ke`None`, skrip akan membuat bucket dan awalan S3 baru. Pastikan ada kebijakan bucket yang sesuai untuk awalan ini.	Tidak	string	
`--enable-instance-protection`	Jika disetel ke`TRUE`, skrip akan membuat kebijakan penghentian kustom (fungsi Lambda) untuk grup Auto Scaling Anda. EC2instance dengan `protected_instance` tag dilindungi dari peristiwa scale-in. Tambahkan `protected_instance` tag ke setiap EC2 instance yang ingin Anda lindungi dari peristiwa scale-in.	Tidak	Boolean	`FALSE`
`--command-logs-bucket`	Nama bucket S3 yang ada untuk menyimpan `AWS ApplyChefRecipe` dan `MountEBSVolumes` log. Jika Anda tidak memberikan nilai, skrip akan membuat bucket S3 baru.	Tidak	string	`aws-opsworks-application-manager-logs-`*`account-id`*

Parameter	Deskripsi	Diperlukan	Tipe	Nilai default
`--custom-json-bucket`	Nama bucket S3 yang ada untuk menyimpan JSON kustom. Jika Anda tidak memberikan nilai, skrip akan membuat bucket S3 baru.	Tidak	string	`aws-apply-chef-application-manager-transition-data-`*`account-id`

Catatan:

- Jika Anda menggunakan GitHub repositori pribadi, Anda harus membuat kunci `Ed25519` host baru untuk SSH. Ini karena GitHub mengubah kunci mana yang didukung di SSH dan menghapus protokol Git yang tidak terenkripsi. Untuk informasi selengkapnya tentang kunci `Ed25519` host, lihat posting GitHub blog Meningkatkan keamanan protokol Git GitHub. Setelah Anda membuat kunci `Ed25519` host baru, buat `SecureString` parameter Systems Manager untuk kunci SSH dan gunakan nama `SecureString` parameter sebagai nilai untuk `--repo-private-key` parameter. Untuk informasi selengkapnya tentang cara membuat `SecureString` parameter Systems Manager, lihat Membuat SecureString parameter (AWS CLI) atau Membuat parameter Systems Manager (konsol) di Panduan AWS Systems Manager Pengguna.
- Parameter`--http-username`, `--http-password` dan `--repo-private-key` parameter mengacu pada nama `SecureString` parameter Systems Manager. Skrip migrasi menggunakan parameter ini saat Anda menjalankan `AWS-ApplyChefRecipes` dokumen.
- `--http-username`Parameter mengharuskan Anda juga menentukan nilai untuk `--http-password` parameter tersebut.
- `--http-password`Parameter mengharuskan Anda juga menentukan nilai untuk `--http-username` parameter tersebut.
- Jangan menetapkan nilai untuk keduanya `--http-password` dan`--repo-private-key`. Berikan nama `SecureString` parameter Systems Manager dari kunci SSH (`--repo-private-key`), atau nama pengguna repositori (`--http-username`) dan kata sandi (). `--http-password`

Langkah 4: Jalankan skrip

Ketika Anda menjalankan`python3 stack_exporter.py`, Anda dapat menyediakan aplikasi, atau membuat template starter dengan menetapkan nilai `--provision-application` parameter ke`FALSE`.

Contoh 1: Menyediakan aplikasi Manajer Aplikasi Systems Manager

Perintah berikut mendapatkan informasi tentang OpsWorks lapisan yang ada, dan menyediakan aplikasi menggunakan OpsWorks arsitektur yang lebih baru, yang mencapai hasil yang mirip dengan versi Chef yang dikonfigurasi untuk tumpukan. Skrip menyediakan semua sumber daya yang diperlukan, seperti grup Auto Scaling dengan menggunakan CloudFormation, dan kemudian mendaftarkan aplikasi di Manajer Aplikasi Systems Manager.

Ganti *`stack-region`* dan *`layer-id`* dengan nilai untuk OpsWorks tumpukan dan layer Anda.

```
python3 stack_exporter.py \
     --layer-id layer-id \
     --region stack-region
```

Contoh 2: Menghasilkan template

Perintah berikut mendapatkan informasi tentang OpsWorks lapisan yang ada dan menghasilkan CloudFormation template. Template, jika disediakan, mencapai hasil yang mirip dengan menggunakan Chef 14. Dalam contoh ini, tidak ada sumber daya yang disediakan, karena `--provision-application` parameter disetel ke. `FALSE`

Ganti *`stack-region`* dan *`layer-id`* dengan nilai untuk OpsWorks tumpukan dan layer Anda.

```
python3 stack_exporter.py \
    --layer-id layer-id \
    --region stack-region \
    --provision-application FALSE
```

Setelah menjalankan perintah, Anda dapat meninjau template di pustaka template Application Manager di Systems Manager, dan Anda juga dapat menyediakan template. Untuk informasi selengkapnya tentang melihat pustaka templat, lihat Bekerja dengan pustaka templat di Panduan AWS Systems Manager Pengguna.

Langkah 5: Menyediakan CloudFormation tumpukan

> **Note**
>
> Anda hanya perlu menyelesaikan langkah ini jika Anda mengatur `--provision-application` parameter untuk skripFALSE.

Saat Anda menentukan `--provision-application` parameter dengan nilaiFALSE, output skrip memberikan nama dan URL untuk CloudFormation template. Template ini merupakan pengganti yang diusulkan untuk OpsWorks tumpukan dan lapisan yang ada.

Anda dapat menyediakan template dengan menggunakan pustaka template Manajer Aplikasi (disarankan), atau dengan menggunakan CloudFormation. Untuk informasi selengkapnya tentang bekerja dengan pustaka templat, lihat Bekerja dengan pustaka templat di Panduan AWS Systems Manager Pengguna.

Langkah 6: Tinjau sumber daya yang disediakan

Anda sekarang siap untuk meninjau sumber daya yang disediakan.

1. Tinjau sumber daya untuk tumpukan yang disediakan menggunakan konsol. AWS CloudFormation
   a. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation dan pilih Stacks.
   b. Pada halaman Stacks, pilih tumpukan, lalu pilih tab Resources.
   c. Pada tab Sumber Daya, tinjau sumber daya yang terdaftar untuk tumpukan Anda. Daftar sumber daya mencakup grup EC2 Auto Scaling, yang dapat Anda tinjau di konsol Auto Scaling, atau. AWS CLI
2. Tinjau sumber daya untuk aplikasi menggunakan Systems Manager Application Manager.
   a. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.
   b. Pada panel navigasi, pilih Manajer Aplikasi.
   c. Di bagian Aplikasi, pilih aplikasi khusus. Manajer Aplikasi membuka tab Gambaran Umum.
   d. Pilih tab Sumber Daya. Tab Resources menampilkan semua sumber daya yang dimigrasikan untuk OpsWorks tumpukan dan lapisan Anda. Nama aplikasi menyertakan nama OpsWorks tumpukan, dan diformat sebagai *app* - *stack-name* - *suffix* di mana

suffix mewakili enam karakter pertama dari ID tumpukan. Untuk informasi selengkapnya tentang melihat sumber daya di Manajer Aplikasi, lihat Melihat sumber daya aplikasi di Panduan AWS Systems Manager Pengguna.

Langkah 7: Mulai sebuah instance

Setelah Anda menyediakan sebuah instance, Anda siap untuk menguji instance tersebut. Pada titik ini, tidak ada contoh yang berjalan.

Untuk membuat instans Anda online, sesuaikan `MinMax`, dan `Desired capacity` nilai untuk grup Auto Scaling ke nomor yang masuk akal untuk aplikasi Anda. Awalnya, Anda mungkin ingin menetapkan nilai ini ke 1, untuk membawa satu instance online, dan memverifikasi instans melakukan semua tindakan yang diharapkan termasuk menjalankan resep Chef kustom Anda.

Langkah 8: Tinjau instance

Setelah Anda memulai sebuah instance, verifikasi itu berjalan seperti yang diharapkan.

1. Tinjau Chef `startup` dan `terminate` log yang terletak di bucket S3 yang ditentukan oleh `--command-logs-bucket` parameter skrip. Secara default, log disimpan dalam ember dengan nama`aws-opsworks-application-manager-logs-`*account-id*.

   a. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   b. Pilih ember yang berisi log Anda.

   c. Arahkan ke `ApplyChefRecipes` awalan untuk melihat log Anda.

2. Periksa konektivitas dan kesehatan Application Load Balancer.

   Ambil langkah-langkah berikut untuk melihat log akses untuk penyeimbang beban Anda. Anda dapat menentukan bucket S3 tempat Anda ingin menyimpan log akses penyeimbang beban dengan menggunakan parameter skrip. `--lb-access-logs-path`

   a. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   b. Pilih bucket S3 Anda, lalu arahkan ke awalan yang berisi log Anda.

3. Verifikasi instans melewati semua pemeriksaan kesehatan Auto Scaling dan Application Load Balancer (jika Anda telah mengonfigurasinya).

Anda dapat melihat informasi tentang kesehatan Auto Scaling di tab Instans baru.

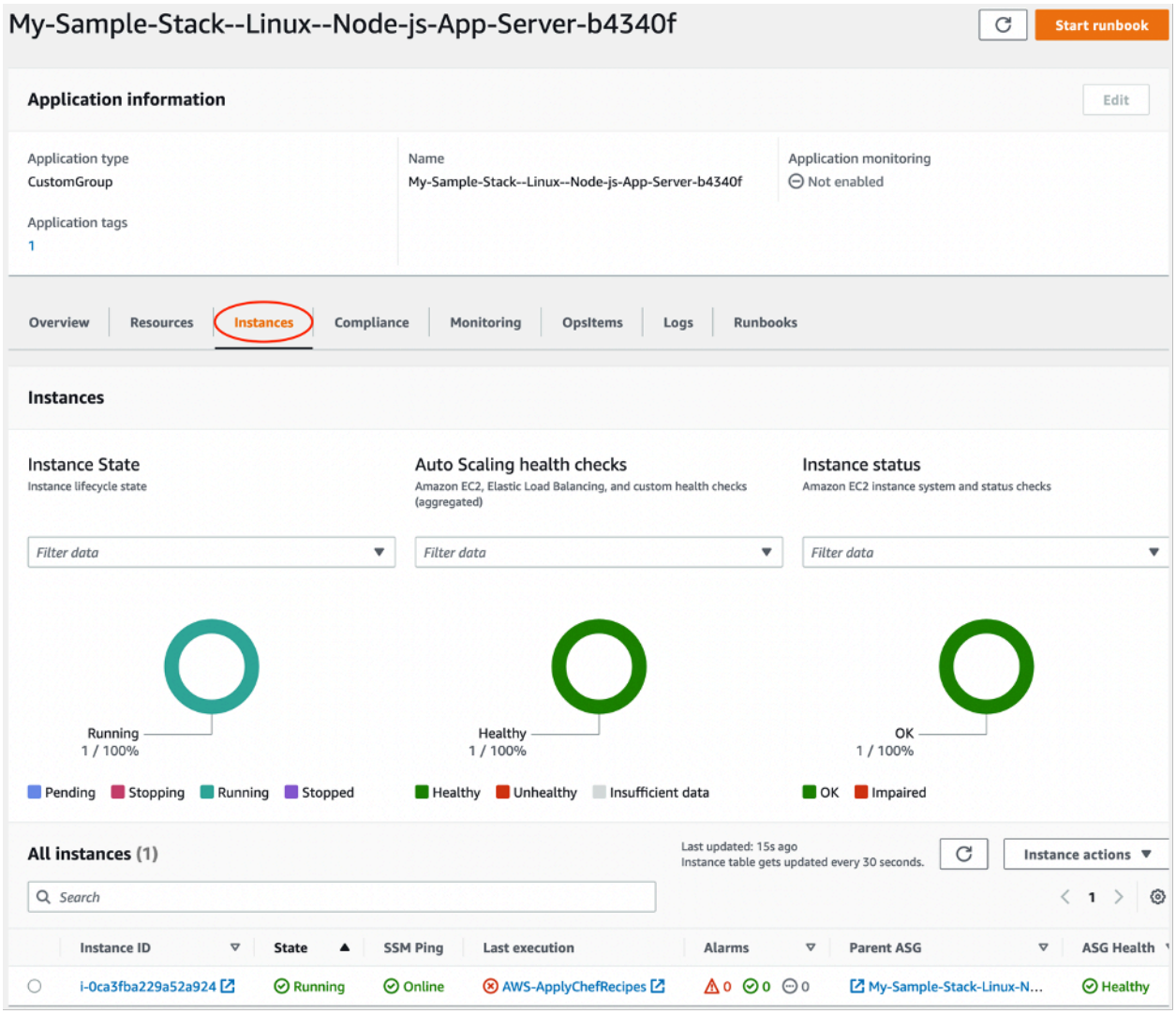
a. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.
b. Pada panel navigasi, pilih Manajer Aplikasi.
c. Di bagian Aplikasi, pilih Aplikasi khusus.
d. Pilih aplikasi dalam daftar. Manajer aplikasi membuka tab Gambaran Umum.
e. Pilih tab Instans untuk melihat informasi tentang kesehatan Auto Scaling.

Setelah memverifikasi bahwa resep Chef berjalan dengan sukses, Anda dapat mengurangi kapasitas grup Auto Scaling untuk menghentikan instance. Jika Anda memiliki resep penghentian khusus, verifikasi resep beroperasi seperti yang diharapkan.

Langkah 9: Memantau dan menjalankan operasi pada instans Anda menggunakan Systems Manager Application Manager

Sekarang Anda dapat memantau dan menjalankan operasi pada instans menggunakan tab Instans baru di halaman Manajer Aplikasi. Untuk informasi selengkapnya tentang bekerja dengan tab Instans, lihat Bekerja dengan instance aplikasi Anda di AWS Systems Manager Panduan Pengguna.

Anda dapat menggunakan tab Instances untuk melihat beberapa AWS instance di satu tempat. Dengan menggunakan tab ini, Anda dapat melihat informasi tentang kesehatan instans dan memecahkan masalah.



Ambil langkah-langkah berikut untuk melihat tab Instans.

1. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.
2. Pada panel navigasi, pilih Manajer Aplikasi.
3. Di bagian Aplikasi, pilih Aplikasi khusus.
4. Pilih aplikasi dalam daftar. Manajer aplikasi membuka tab Gambaran Umum.
5. Pilih tab Instans untuk melihat informasi tentang status instans dan EC2 kesehatan Anda.

Pertanyaan yang Sering Diajukan

Berikut ini FAQs memberikan jawaban atas beberapa pertanyaan umum.

Topik

- AWS OpsWorks Stacks Versi mana yang dapat saya migrasi?
- Versi Chef mana yang dapat digunakan instance migrasi saya?
- Jenis repositori mana yang dapat saya migrasikan?
- Dapatkah saya terus menggunakan repositori Git pribadi?
- Kunci SSH apa yang dapat saya gunakan untuk mengakses instans saya?
- Mengapa instans saya secara otomatis masuk dan keluar?
- Bisakah saya mematikan Auto Scaling?
- Dapatkah saya melakukan pembaruan kernel dan paket pada EC2 instance yang diluncurkan?
- Mengapa volume EBS dalam instans saya tidak berisi data apa pun?
- Mengapa volume EBS yang dijelaskan dalam template peluncuran saya tidak dipasang?
- Di mana saya dapat menemukan resep Chef dan log volume Mount EBS?
- Di mana saya dapat menemukan log debug untuk skrip migrasi?
- Apakah skrip migrasi mendukung pembuatan versi CloudFormation templat?
- Bisakah saya memigrasi beberapa lapisan?
- Bagaimana cara membuat SecureString parameter?
- Bagaimana cara melindungi instans di grup Auto Scaling baru dari peristiwa penghentian?
- Penyeimbang beban apa yang tersedia dengan skrip migrasi?
- Apakah resep konfigurasi buku masak khusus dimigrasikan?
- Bisakah saya menjalankan resep deploy dan undeploy pada instance yang baru saya buat?
- Dapatkah saya mengubah subnet apa yang terbentang oleh grup Auto Scaling saya?

AWS OpsWorks Stacks Versi mana yang dapat saya migrasi?

Anda hanya dapat memigrasikan tumpukan Chef 11.10 dan Chef 12, Amazon Linux, Amazon Linux 2, Ubuntu, dan Red Hat Enterprise Linux 7.

Versi Chef mana yang dapat digunakan instance migrasi saya?

Instans yang dimigrasi dapat menggunakan Chef versi 11 hingga 14.

> **Note**
>
> Migrasi tumpukan Windows tidak didukung.

Jenis repositori mana yang dapat saya migrasikan?

Anda dapat memigrasikan tipe repositori S3, Git, dan HTTP.

Dapatkah saya terus menggunakan repositori Git pribadi?

Ya, Anda dapat terus menggunakan repositori Git pribadi.

Jika Anda menggunakan GitHub repositori pribadi, Anda harus membuat kunci `Ed25519` host baru untuk SSH. Ini karena GitHub mengubah kunci mana yang didukung di SSH dan menghapus protokol Git yang tidak terenkripsi. Untuk informasi selengkapnya tentang kunci `Ed25519` host, lihat posting GitHub blog Meningkatkan keamanan protokol Git GitHub. Setelah Anda membuat kunci `Ed25519` host baru, buat `SecureString` parameter Systems Manager untuk kunci SSH ini dan gunakan nama parameter sebagai nilai untuk `--repo-private-key` parameter. Untuk informasi selengkapnya tentang cara membuat `SecureString` parameter Systems Manager, lihat Membuat SecureString parameter (AWS CLI) di Panduan AWS Systems Manager Pengguna.

Untuk jenis repositori Git lainnya, buat `SecureString` parameter Systems Manager untuk kunci SSH ini dan gunakan nama parameter sebagai nilai untuk parameter skrip. `--repo-private-key`

Kunci SSH apa yang dapat saya gunakan untuk mengakses instans saya?

Saat Anda menjalankan skrip, skrip memigrasikan kunci SSH dan instance yang dikonfigurasi dalam tumpukan. Anda dapat menggunakan kunci SSH untuk mengakses instans Anda. Jika kunci SSH disediakan untuk tumpukan dan instance, skrip menggunakan kunci dari tumpukan. Jika Anda tidak yakin kunci SSH mana yang akan digunakan, lihat instance di EC2 console () https://console.aws.amazon.com/ec2/. Halaman Detail di EC2 konsol menampilkan kunci SSH untuk instans Anda.

Mengapa instans saya secara otomatis masuk dan keluar?

Auto Scaling menskalakan instans berdasarkan aturan penskalaan untuk grup Auto Scaling. Anda dapat mengatur nilai kapasitas Min, Maks, dan Diinginkan untuk grup Anda. Grup Auto Scaling secara otomatis menskalakan kapasitas Anda saat Anda memperbarui nilai ini.

Bisakah saya mematikan Auto Scaling?

Anda dapat menonaktifkan Auto Scaling dengan menyetel nilai kapasitas Min, Maks, dan Diinginkan grup Auto Scaling ke nomor yang sama. Misalnya, jika Anda ingin selalu memiliki sepuluh instance, atur nilai kapasitas Min, Max, dan Diinginkan menjadi sepuluh.

Dapatkah saya melakukan pembaruan kernel dan paket pada EC2 instance yang diluncurkan?

Secara default, pembaruan kernel dan paket terjadi saat EC2 instance boot. Gunakan langkah-langkah berikut untuk melakukan pembaruan kernel atau paket pada EC2 instance yang diluncurkan. Misalnya, Anda mungkin ingin menerapkan pembaruan setelah menjalankan penerapan atau konfigurasi resep.

1. Connect ke EC2 instans Anda.
2. Buat `perform_upgrade` fungsi berikut dan jalankan pada instance Anda.

```
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. Setelah pembaruan kernel dan paket, Anda mungkin perlu me-reboot EC2 instance Anda. Untuk memeriksa apakah Anda reboot diperlukan, buat `reboot_if_required` fungsi berikut dan jalankan pada EC2 instance Anda.

```
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. Jika menjalankan `reboot_if_required` hasil dalam `reboot is required` pesan, reboot EC2 instance. Jika Anda menerima `reboot is not required` pesan, Anda tidak perlu me-reboot EC2 instance.

Mengapa volume EBS dalam instans saya tidak berisi data apa pun?

Saat Anda menjalankan skrip, skrip memigrasikan konfigurasi volume EBS, menciptakan arsitektur pengganti untuk OpsWorks tumpukan dan lapisan Anda. Skrip tidak memigrasikan instance aktual atau data yang terkandung dalam instance. Skrip hanya memigrasikan konfigurasi volume EBS pada tingkat lapisan dan melampirkan volume EBS kosong ke instance yang diluncurkan. EC2

Ambil langkah-langkah berikut untuk menarik data dari volume EBS instans sebelumnya.

1. Ambil snapshot dari volume EBS instans Anda sebelumnya. Untuk informasi selengkapnya tentang membuat snapshot, lihat Membuat snapshot Amazon EBS di Panduan Pengguna Amazon EC2 .
2. Buat volume dari snapshot Anda. Untuk informasi selengkapnya tentang membuat volume dari snapshot, lihat Membuat volume dari snapshot di EC2 Panduan Pengguna Amazon.
3. Lampirkan volume yang Anda buat ke instance. Untuk informasi selengkapnya tentang melampirkan volume, lihat Melampirkan volume Amazon EBS ke instans di EC2 Panduan Pengguna Amazon.

Mengapa volume EBS yang dijelaskan dalam template peluncuran saya tidak dipasang?

Jika Anda memberikan ID template peluncuran untuk `--launch-template` parameter dengan volume EBS, skrip melampirkan volume EBS, tetapi tidak memasang volume. Anda dapat memasang volume EBS terlampir dengan menjalankan `MountEBSVolumes` RunCommand dokumen yang dibuat skrip untuk EC2 instance yang diluncurkan.

Jika Anda tidak mengatur `--launch-template` parameter, skrip akan membuat template, dan ketika grup Auto Scaling meluncurkan EC2 instance baru, grup Auto Scaling secara otomatis melampirkan volume EBS dan kemudian menjalankan `SetupAutomation` perintah untuk memasang volume yang terpasang ke titik pemasangan yang dikonfigurasi dalam pengaturan lapisan.

Di mana saya dapat menemukan resep Chef dan log volume Mount EBS?

OpsWorks mengirimkan log ke bucket S3 yang dapat Anda tentukan dengan memberikan nilai untuk parameter. `--command-logs-bucket` Nama bucket S3 default memiliki format:`aws-opsworks-stacks-application-manager-logs-`*`account-id`*. Log resep koki disimpan di `ApplyChefRecipes` awalan. Log volume Mount EBS disimpan dalam `MountEBSVolumes` awalan. Semua layer yang dimigrasikan dari tumpukan mengirimkan log ke bucket S3 yang sama.

> **Note**
>
> - Konfigurasi Siklus Hidup bucket S3 menyertakan aturan untuk menghapus log setelah 30 hari. Jika ingin menyimpan log selama lebih dari 30 hari, Anda harus memperbarui aturan dalam konfigurasi Siklus Hidup bucket S3.
> - Saat ini, OpsWorks hanya log Chef `setup` dan `terminate` resep.

Di mana saya dapat menemukan log debug untuk skrip migrasi?

Skrip menempatkan log debug dalam ember bernama`aws-opsworks-stacks-transition-logs-`*`account-id`*. Anda dapat menemukan log debug di `migration_script` folder bucket S3 di bawah folder yang cocok dengan nama layer yang Anda migrasi.

Apakah skrip migrasi mendukung pembuatan versi CloudFormation templat?

Skrip menghasilkan dokumen Systems Manager dari jenis CloudFormation yang membuat pengganti lapisan atau tumpukan yang ingin Anda migrasikan. Menjalankan skrip lagi, bahkan dengan parameter yang sama, mengekspor versi baru dari template layer yang diekspor sebelumnya. Versi template disimpan dalam bucket S3 yang sama dengan log skrip.

Bisakah saya memigrasi beberapa lapisan?

`--layer-id`Parameter skrip lewat dalam satu lapisan. Untuk memigrasikan beberapa lapisan, jalankan kembali skrip dan teruskan yang berbeda. `--layer-id`

Lapisan yang merupakan bagian dari OpsWorks tumpukan yang sama terdaftar di bawah aplikasi yang sama di Application Manager.

1. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.
2. Pada panel navigasi, pilih Manajer Aplikasi.
3. Di bagian Aplikasi, pilih Aplikasi khusus.
4. Pilih aplikasi Anda. Nama aplikasi dimulai dengan`app-`*`stack-name`*`-`*`first-six-characters-stack-id`*.
5. Elemen tingkat atas yang dimulai dengan aplikasi, menampilkan semua komponen yang sesuai dengan OpsWorks tumpukan Anda. Ini termasuk komponen yang sesuai dengan OpsWorks lapisan Anda.
6. Pilih komponen yang sesuai dengan layer untuk melihat sumber daya untuk layer. Komponen yang mewakili OpsWorks lapisan juga terlihat dari bagian aplikasi Kustom sebagai aplikasi individual.

Bagaimana cara membuat **`SecureString`** parameter?

Anda dapat menggunakan Systems Manager untuk membuat `SecureString` parameter. Untuk informasi selengkapnya tentang cara membuat `SecureString` parameter Systems Manager, lihat Membuat SecureString parameter (AWS CLI) atau Membuat parameter Systems Manager (konsol) di Panduan AWS Systems Manager Pengguna.

Anda harus memberikan `SecureString` parameter sebagai nilai untuk`--http-username`,`--http-password`, atau `--repo-private-key` parameter.

Bagaimana cara melindungi instans di grup Auto Scaling baru dari peristiwa penghentian?

Anda dapat melindungi instance dengan menyetel `--enable-instance-protection` parameter `TRUE` dan menambahkan kunci `protected_instance` tag ke setiap EC2 instance yang ingin Anda lindungi dari peristiwa penghentian. Saat Anda menyetel `--enable-instance-protection` parameter `TRUE` dan menambahkan kunci `protected_instance` tag, skrip akan menambahkan kebijakan penghentian kustom ke grup Auto Scaling baru Anda dan menangguhkan prosesnya. `ReplaceUnhealthy` Instans dengan kunci `protected_instance` tag dilindungi dari peristiwa penghentian berikut:

- Skala dalam acara

- Refresh instans
- Menyeimbangkan kembali
- Instance max seumur hidup
- Izinkan pengakhiran instans daftar
- Pengakhiran dan penggantian contoh yang tidak sehat

> **Note**
>
> Anda harus mengatur kunci `protected_instance` tag pada instance yang ingin Anda lindungi. Kunci tag peka huruf besar/kecil. Setiap instance dengan kunci tag itu dilindungi terlepas dari nilai tag.
> Untuk mengurangi waktu proses kebijakan penghentian kustom, Anda dapat meningkatkan jumlah instans default yang digunakan fungsi Lambda untuk memfilter instance yang dilindungi dengan memperbarui nilai variabel kode fungsi. `default_sample_size` Nilai defaultnya adalah 15. Jika Anda meningkatkan`default_sample_size`, Anda mungkin perlu meningkatkan memori yang dialokasikan ke fungsi Lambda, yang akan meningkatkan biaya fungsi Lambda Anda. Untuk informasi selengkapnya tentang harga AWS Lambda , lihat Harga AWS Lambda.

Penyeimbang beban apa yang tersedia dengan skrip migrasi?

Skrip ini menyediakan tiga opsi penyeimbang beban.

- (Disarankan) Buat Application Load Balancer baru. Secara default, skrip membuat Application Load Balancer baru. Anda juga dapat mengatur `--lb-type` parameter ke`ALB`. Untuk informasi selengkapnya tentang Application Load Balancer, lihat Apa itu Application Load Balancer? dalam Panduan Pengguna Elastic Load Balancing.
- Jika Application Load Balancer bukan pilihan, buat Classic Load Balancer dengan menyetel `--lb-type` parameter ke. `Classic` Jika Anda memilih opsi ini, Classic Load Balancer yang ada yang terpasang pada OpsWorks layer Anda akan tetap terpisah dari aplikasi Anda. Untuk informasi selengkapnya tentang Application Load Balancer, lihat Apa itu Classic Load Balancer? dalam Panduan Pengguna Elastic Load Balancing: Classic Load Balancers.
- Anda dapat melampirkan penyeimbang beban yang ada dengan menyetel `--lb-type` parameter ke`None`.

> **Important**
>
> Sebaiknya buat load balancer Elastic Load Balancing baru untuk layer AWS Stacks Anda. OpsWorks Jika Anda memilih untuk menggunakan penyeimbang beban Elastic Load Balancing yang ada, Anda harus terlebih dahulu mengonfirmasi bahwa itu tidak digunakan untuk tujuan lain dan tidak memiliki instans yang terpasang. Setelah load balancer dilampirkan ke layer, OpsWorks menghapus instance yang ada dan mengonfigurasi load balancer untuk menangani hanya instance layer. Meskipun secara teknis dimungkinkan untuk menggunakan konsol Elastic Load Balancing atau API untuk memodifikasi konfigurasi load balancer setelah melampirkannya ke lapisan, Anda tidak boleh melakukannya; perubahan tidak akan permanen.

Untuk melampirkan penyeimbang beban OpsWorks layer yang ada ke grup Auto Scaling

1. Jalankan skrip migrasi dengan `--lb-type` parameter yang disetel ke`None`. Ketika nilai diatur ke`None`, skrip tidak mengkloning atau membuat penyeimbang beban.
2. Setelah skrip menyebarkan CloudFormation tumpukan, perbarui `Min` `Max` grup `Desired capacity` dan nilai Auto Scaling, lalu uji aplikasi Anda.
3. Pilih yang `Link to the template` ditampilkan dalam output skrip. Jika Anda menutup terminal Anda, ambil langkah-langkah ini untuk mengakses template.
   a. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.
   b. Pada panel navigasi, pilih Manajer Aplikasi.
   c. Pilih CloudFormation tumpukan dan kemudian pilih pustaka Template.
   d. Pilih Dimiliki oleh saya dan temukan template Anda.
4. Dari CloudFormation template, pilih Edit dari menu Tindakan.
5. Perbarui `LabelBalancerNames` properti dalam bagian `ApplicationAsg` sumber daya CloudFormation template.

```
ApplicationAsg:
   DependsOn: CustomTerminationLambdaPermission
   Properties:
   #(other properties in ApplicationAsg to remain unchanged)
      LoadBalancerNames:
        - load-balancer-name
```

```
      HealthCheckType: ELB
```

6. Jika Anda ingin pemeriksaan kesehatan instans grup Auto Scaling Anda juga menggunakan pemeriksaan kesehatan penyeimbang beban, hapus bagian di bawah ini dan masukkan. `HealthCheckType ELB` Jika Anda hanya memerlukan pemeriksaan EC2 kesehatan, Anda tidak perlu mengubah template.
7. Simpan perubahan Anda. Menyimpan membuat versi default baru dari template. Jika ini adalah pertama kalinya Anda menjalankan skrip untuk lapisan dan pertama kali Anda menyimpan perubahan di konsol, versi yang lebih baru adalah 2.
8. Dari Actions, pilih Provision stack.
9. Konfirmasikan bahwa Anda ingin menggunakan versi default template. Pastikan Pilih tumpukan yang ada dipilih dan pilih CloudFormation tumpukan yang akan diperbarui.
10. Pilih Berikutnya untuk setiap halaman berikutnya hingga Anda melihat halaman Tinjauan dan Ketentuan. Pada halaman Tinjauan dan Ketentuan, pilih keduanya yang saya akui yang AWS CloudFormation mungkin membuat sumber daya IAM dengan nama khusus dan saya memahami bahwa perubahan pada templat yang dipilih dapat AWS CloudFormation menyebabkan pembaruan atau penghapusan sumber daya yang ada AWS .
11. Pilih tumpukan Provision.

Jika Anda perlu memutar kembali pembaruan Anda, lakukan langkah-langkah berikut.

1. Pilih Actions dan kemudian pilih Provision stack.
2. Pilih salah satu versi yang ada dan kemudian pilih versi template sebelumnya.
3. Pilih Pilih tumpukan yang ada dan kemudian pilih CloudFormation tumpukan yang akan diperbarui.

Apakah resep konfigurasi buku masak khusus dimigrasikan?

Konfigurasi buku masak khusus tidak didukung untuk dijalankan selama acara penyiapan. Skrip memigrasikan resep konfigurasi buku masak kustom dan membuat runbook Otomasi Systems Manager untuk Anda. Namun, Anda harus menjalankan resep secara manual.

Ambil langkah-langkah berikut untuk menjalankan resep konfigurasi Anda.

1. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.

2. Pada panel navigasi, pilih Manajer Aplikasi.
3. Di bagian Aplikasi, pilih Aplikasi khusus.
4. Pilih aplikasi Anda. Nama aplikasi dimulai dengan`app-`*`stack-name`*.
5. Pilih Resources dan kemudian pilih runbook konfigurasi.
6. Pilih Jalankan Otomasi.
7. Pilih IDs contoh yang ingin Anda jalankan resep konfigurasi dan kemudian pilih Execute.

Bisakah saya menjalankan resep deploy dan undeploy pada instance yang baru saya buat?

Skrip dapat membuat tiga kemungkinan runbook Otomasi tergantung pada konfigurasi lapisan Anda.

- Pengaturan
- Konfigurasi
- Mengakhiri

Script juga dapat membuat parameter Systems Manager berikut yang berisi nilai input untuk `AWS-ApplyChefRecipes Run Command` dokumen.

- Pengaturan
- Deploy
- Konfigurasi
- Undeploy
- Mengakhiri

Ketika peristiwa scale-out terjadi, runbook Automation setup berjalan secara otomatis. Ini termasuk pengaturan dan penerapan resep buku masak khusus dari lapisan asli OpsWorks Anda. Ketika peristiwa scale-in terjadi, runbook Otomasi terminate berjalan secara otomatis. Runbook terminate Automation berisi resep shutdown dari layer asli Anda. OpsWorks

Jika Anda ingin menjalankan undeploy atau mengkonfigurasi resep secara manual, lakukan langkah-langkah berikut.

1. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.
2. Pada panel navigasi, pilih Manajer Aplikasi.

3. Di bagian Aplikasi, pilih Aplikasi khusus.
4. Pilih aplikasi Anda. Nama aplikasi dimulai dengan`app-`*`stack-name`*`-`*`first-six-characters-stack-id`*. Manajer aplikasi membuka tab Gambaran Umum.
5. Pilih Sumber Daya dan kemudian pilih runbook konfigurasi Otomasi.
6. Pilih Jalankan Otomasi.
7. Untuk parameter input Runbook `applyChefRecipesPropertiesParameter` Otomasi, rujuk parameter Systems Manager yang benar. Nama parameter Systems Manager mengikuti format`/ApplyChefRecipes-Preset/`*`OpsWorks-stack-name`*`-`*`OpsWorks-layer-name`*`-`*`first-six-characters-stack-id`*`/`*`event`*, di mana nilainya `ConfigureDeploy`, atau `Undeploy` tergantung pada resep yang ingin Anda jalankan. *`event`*
8. Pilih contoh di IDs mana Anda ingin menjalankan resep dan pilih Execute.

Dapatkah saya mengubah subnet apa yang terbentang oleh grup Auto Scaling saya?

Secara default, grup Auto Scaling mencakup semua subnet di VPC stack Anda. OpsWorks Untuk memperbarui subnet mana yang akan direntang, lakukan langkah-langkah berikut.

1. Pilih yang `Link to the template` ditampilkan dalam output skrip. Jika Anda menutup terminal Anda, ambil langkah-langkah ini untuk mengakses template.
   a. Buka konsol Systems Manager di https://console.aws.amazon.com/systems-manager/.
   b. Pada panel navigasi, pilih Manajer Aplikasi.
   c. Pilih CloudFormation tumpukan dan kemudian pilih pustaka Template.
   d. Pilih Dimiliki oleh saya dan temukan template Anda.
2. Dari Actions, pilih Provision stack.
3. Konfirmasikan bahwa Anda ingin menggunakan templat default. Pilih Pilih tumpukan yang ada dan kemudian pilih CloudFormation tumpukan yang akan diperbarui.

   > **Note**
   >
   > Jika Anda menjalankan skrip dengan `--provision-application` parameter yang disetel ke`FALSE`, Anda harus membuat CloudFormation tumpukan baru.

4. Untuk `SubnetIDs` parameternya, berikan daftar subnet yang dipisahkan koma IDs yang Anda inginkan untuk menjangkau grup Auto Scaling Anda.

5. Pilih Berikutnya hingga Anda melihat halaman Tinjauan dan Ketentuan.
6. Pada halaman Tinjauan dan Penyediaan, pilih Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM dengan nama khusus dan saya memahami bahwa perubahan pada templat yang dipilih dapat AWS CloudFormation menyebabkan pembaruan atau penghapusan sumber daya yang ada AWS .
7. Pilih tumpukan Provision.

Pemecahan Masalah

Bagian ini berisi beberapa masalah umum, dan solusi yang disarankan untuk masalah tersebut.

Topik

- Prinsipal yang diberikan tidak valid
- Tidak dapat menghapus CloudFormation tumpukan saat instance yang dilindungi grup Auto Scaling diaktifkan
- Kesalahan akses ditolak saat menyediakan bucket dan awalan S3 yang ada

Prinsipal yang diberikan tidak valid

Masalah: Anda menerima pesan kesalahan yang menyatakan bahwa prinsipal yang Anda berikan tidak valid.

Penyebab: Ini terjadi karena grup Auto Scaling tidak memiliki peran layanan.

Solusi: Buat grup Auto Scaling di Wilayah tempat kesalahan terjadi. Membuat grup Auto Scaling akan menciptakan peran terkait layanan yang diperlukan untuk kebijakan penghentian kustom Anda.

Tidak dapat menghapus CloudFormation tumpukan saat instance yang dilindungi grup Auto Scaling diaktifkan

Masalah: `--enable-instance-protection` Parameter disetel ke `TRUE` dan beberapa EC2 instance grup Auto Scaling Anda dilindungi dengan kunci `protected_instance` tag, yang mencegah AWS CloudFormation tumpukan Anda dihapus sepenuhnya.

Penyebab: EC2 Instance memiliki kunci `protected_instance` tag yang melindunginya dari peristiwa penghentian.

Solusi: Hapus kunci `protected_instance` tag dari EC2 instance. Hal ini memungkinkan grup Auto Scaling untuk menurunkan skala. Setelah grup Auto Scaling menurunkan skala, Anda dapat menghapus tumpukan. AWS CloudFormation

Kesalahan akses ditolak saat menyediakan bucket dan awalan S3 yang ada

Masalah: Anda menerima `AccessDenied` kesalahan saat memberikan bucket dan awalan S3 yang ada.

Penyebab: Kebijakan bucket S3 tidak memberikan izin yang diperlukan untuk mengirimkan log penyeimbang beban ke bucket.

Solusi: Perbarui kebijakan bucket S3 untuk memungkinkan skrip mengirimkan log akses penyeimbang beban ke bucket. Untuk informasi selengkapnya tentang cara memperbarui kebijakan bucket, lihat Mengaktifkan log akses untuk Application Load Balancer di Panduan Pengguna Elastic Load Balancing: Application Load Balancers.

Menggunakan alat AWS OpsWorks Stacks Detach in Place

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin.

Bagian ini menjelaskan cara menggunakan alat AWS OpsWorks Stacks Lepaskan di Tempat untuk melepaskan OpsWorks instance Anda dari layanan Stacks. OpsWorks

Contoh yang Anda lepaskan akan tetap ada di Anda Akun AWS, tetapi Anda tidak akan lagi dapat mengelolanya menggunakan. OpsWorks Sebagai gantinya, Anda akan menggunakan Amazon EC2, AWS Systems Manager, atau pendekatan apa pun yang EC2 kompatibel untuk mengonfigurasi dan mengelola instance.

Pada tingkat tinggi, proses detasemen melibatkan langkah-langkah berikut:

1. Alat ini melakukan pemeriksaan validasi untuk memastikan sumber daya siap untuk dilepas.
2. Alat ini mengekspor JSON Kustom dari OpsWorks tumpukan Anda dan menyimpannya sebagai objek di Amazon S3.

3. Alat ini membuat dokumen Otomasi Systems Manager yang mewakili setiap peristiwa siklus hidup OpsWorks Stacks.
4. Alat ini membuat AWS Service Catalog AppRegistry Katalog untuk semua instance yang sedang dilepas dan melepaskan penyeimbang beban Elastic Load Balancing (ELB) dari lapisan. OpsWorks
5. Terakhir, alat ini melepaskan dan membatalkan pendaftaran sumber daya lain termasuk instans Amazon Relational Database Service (Amazon RDS).

Bagaimana prosesnya bekerja

Alat Detach In Place menyediakan 3 perintah berikut dan pengalaman seperti wizard yang memandu Anda melalui serangkaian langkah untuk memeriksa dan mengonfigurasi instance Anda sebelum melanjutkan untuk melepaskan lapisan Anda.

Perintah	Deskripsi
`handle-prerequisites`	Perintah ini menganalisis apakah semua instance dalam lapisan memenuhi syarat untuk detasemen dan menyelesaikan prasyarat. Instans harus dalam keadaan sehat OpsWorks, mereka tidak dapat memiliki penskalaan otomatis berbasis waktu atau beban, dan harus menginstal versi OpsWorks Agen terbaru. Selain itu, perintah memeriksa apakah semua instance memiliki izin yang diperlukan untuk mendukung Agen SSM dan apakah versi Agen SSM terbaru diinstal. Perintah akan menginstal Agen SSM jika tidak ada dan akan memperbar ui Agen SSM jika tidak menggunakan versi terbaru. Perintah juga akan menambahkan izin yang diperlukan.
`detach`	Perintah ini melepaskan semua OpsWorks instance untuk lapisan yang ditentukan.

Perintah	Deskripsi
	Pertama, perintah akan menjalankan pemeriksaan prasyarat untuk memastikan lapisan memenuhi syarat untuk detasemen. Jika Anda tidak ingin menyelesaikan prasyarat, Anda diberi opsi untuk melepaskan paksa. Selanjutnya, perintah akan menunjukkan bahwa semua tag yang ditambahkan ke instance Anda melalui OpsWorks penandaan APIs atau melalui propagasi tag dari lapisan dan tumpukan Anda, akan dipertahankan. Anda dapat menghapus salah satu tag ini menggunakan relevan EC2 APIs setelah detasemen selesai. Kemudian, perintah akan memeriksa apakah Anda ingin mengekspor konfigurasi terkait Chef ke parameter SSM. Jika Anda memiliki Classic Load Balancer yang terpasang pada layer, perintah akan menanyakan apakah itu dapat melepaska n penyeimbang beban untuk mencegah downtime.

Perintah	Deskripsi
`cleanup`	Perintah ini menghapus semua entitas OpsWorks dari akun Anda. Ini akan mengakhiri instance dan menghapus semua tumpukan. Ini harus digunakan untuk sumber daya yang tidak lagi diperlukan sebagai langkah terakhir untuk membersihkan akun. **Note** Kami menyarankan Anda menjalankan pengaturan baru selama beberapa hari sebelum menjalankan `cleanup` perintah. Ini memastikan bahwa setiap konfigurasi yang diperlukan dari tumpukan sudah tersedia jika diperlukan.

Batasan

Tujuan utama alat Detach In Place adalah untuk melepaskan instance OpsWorks Stacks dengan aman. Bagian ini merangkum keterbatasan alat.

- Agen SSM Windows - Jika Agen SSM tidak diinstal pada instance, Anda harus menginstalnya secara manual. Hal yang sama berlaku jika Agen tidak diperbarui ke versi terbaru.
- Instans Time/Load Auto Scaling - Alat detasemen tidak mendukung instance dengan Auto Scaling diaktifkan. Anda harus menonaktifkan Auto Scaling pada instance yang ingin Anda lepaskan.
- Izin — Alat detasemen tidak membuat atau menghasilkan entitas IAM yang ditentukan pada halaman Izin konsol. OpsWorks
- Aplikasi — Alat detasemen tidak membuat atau menghasilkan aplikasi di luar. OpsWorks

Memulai

Langkah 1: Verifikasi prasyarat terpenuhi

Semua 3 perintah dari alat Detach In Place adalah skrip Python, yang dapat Anda jalankan secara lokal, misalnya, EC2 atau dengan menggunakan. AWS CloudShell

AWS CloudShell adalah shell berbasis browser yang memberi Anda akses baris perintah ke AWS sumber daya yang dipilih. Wilayah AWS AWS CloudShell datang pra-instal dengan alat-alat populer (seperti AWS CLI dan Python). Saat menggunakan AWS CloudShell, Anda menggunakan kredensyal yang sama dengan yang Anda gunakan untuk masuk ke konsol.

Panduan ini mengasumsikan bahwa Anda menggunakan. AWS CloudShell

Langkah 2: Unduh skrip

1. Unduh file zip yang berisi skrip migrasi dan semua file yang relevan dengan menjalankan perintah berikut:

   ```
   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. Unzip file dengan menjalankan perintah berikut.

   ```
   unzip detach_in_place_script.zip
   ```

   Setelah file di-unzip, file berikut tersedia:

   - README.md
   - LISENSI
   - MELIHAT
   - requirements.txt
   - TODO.py

3. Jika perlu, instal `pipenv` dengan menjalankan perintah berikut.

   ```
   pip install pipenv
   ```

Langkah 3: Jalankan skrip

Pertama, atur lingkungan Anda sehingga Anda dapat menjalankan skrip dengan menjalankan perintah berikut.

```
pipenv install -r requirements.txt
pipenv shell
```

Kemudian, tinjau parameter skrip.

Perintah	Parameter	Deskripsi	Jenis	Diperlukan	Default
`handle-prerequisites`	`--layer-id`	ID dari layer yang ingin Anda lepaskan.	Tali	Ya	-
	`--region`	Wilayah OpsWorks tumpukan. Jika OpsWorks Stack Region dan API endpoint Region berbeda, gunakan Stack Region. Ini adalah Wilayah yang sama dengan bagian sumber daya lain dari OpsWorks tumpukan Anda (misalnya, EC2 instance dan subnet).	String	Tidak	us-east-1
`detach`	`--layer-id`	ID dari layer yang ingin Anda lepaskan.	Tali	Ya	-
	`--batch-size`	Jumlah contoh untuk melepaskan dari lapisan (misalnya, 5).	String	Tidak	-
	`--region`	Wilayah OpsWorks tumpukan. Jika OpsWorks Stack Region dan API endpoint Region berbeda,	String	Tidak	us-east-1

Perintah	Parameter	Deskripsi	Jenis	Diperlukan	Default
		gunakan Stack Region. Ini adalah Wilayah yang sama dengan bagian sumber daya lain dari OpsWorks tumpukan Anda (misalnya, EC2 instance dan subnet).			
`cleanup`	`--stack-id`	ID tumpukan yang ingin Anda hapus.	String	Tidak	Saling eksklusif, Anda harus menentukan baik ID lapisan atau ID tumpukan
	`--layer-id`	ID dari layer yang ingin Anda hapus	String	Tidak	
	`--region`	Wilayah OpsWorks tumpukan. Jika OpsWorks Stack Region dan API endpoint Region berbeda, gunakan Stack Region. Ini adalah Wilayah yang sama dengan bagian sumber daya lain dari OpsWorks tumpukan Anda (misalnya, EC2 instance dan subnet).	String	Tidak	us-east-1

Anda dapat melihat opsi yang tersedia untuk`detach`, `handle-prerequisites` dan `cleanup` perintah dengan menjalankan perintah dengan `--help` opsi sebagai berikut:

```
python3 layer_detacher.py detach --help
python3 layer_detacher.py handle-prerequisites --help
python3 layer_detacher.py cleanup --help
```

Anda sekarang siap untuk memulai. Contoh berikut menunjukkan bagaimana Anda dapat menjalankan perintah untuk kasus penggunaan yang berbeda.

Contoh:

- Contoh 1: Periksa apakah lapisan memenuhi semua prasyarat dan memenuhi syarat untuk detasemen
- Contoh 2: Lepaskan semua instance layer
- Contoh 3: Lepaskan semua instance layer dalam batch
- Contoh 4: Bersihkan semua sumber daya untuk lapisan dan hapus layer
- Contoh 5: Bersihkan semua sumber daya untuk tumpukan dan hapus tumpukan

Contoh 1: Periksa apakah lapisan memenuhi semua prasyarat dan memenuhi syarat untuk detasemen

Perintah berikut membaca informasi tentang OpsWorks lapisan (dan instance yang disertakan) dan memeriksa apakah prasyarat berikut terpenuhi:

- Semua contoh online.
- Tidak ada instans Load/Time Auto Scaling.
- Semua contoh memiliki OpsWorks Agen terbaru.
- Semua instance memiliki Agen SSM terbaru yang diinstal dan dikonfigurasi.
- Semua instance memiliki key pair SSH.
- Setiap contoh milik tepat satu lapisan.

```
python3 layer_detacher.py handle-prerequisites \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Contoh 2: Lepaskan semua instance layer

Perintah berikut akan mengulangi semua instance layer, memeriksa apakah instance memenuhi prasyarat dan mencoba melepaskan secara paralel semua instance yang memenuhi prasyarat. Jika satu atau lebih prasyarat tidak terpenuhi, perintah akan memberikan opsi pelepasan paksa untuk instance yang tidak sesuai yang tersisa.

Sebelum melepaskan instance apa pun, perintah akan:

1. Simpan JSON Kustom dan unggah ke S3.
2. Buat dokumen SSM Automation untuk setiap peristiwa OpsWorks siklus hidup untuk layer dan unggah log eksekusi untuk dokumen Automation ke S3.
3. Buat AppRegistry aplikasi untuk semua instance yang akan terlepas. Aplikasi ini memiliki Grup Sumber Daya yang terkait dengannya yang menyimpan semua instance dan sumber daya yang terpisah. Sumber daya termasuk dokumen Otomasi SSM dan parameter SSM yang menyimpan informasi tentang peristiwa siklus hidup dan resep Chef khusus.
4. Melepaskan Classic Load Balancer dari layer, jika ada.

Perintah ini hanya akan memodifikasi OpsWorks sumber daya. Status EC2 instance akan tetap sama.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Contoh 3: Lepaskan semua instance layer dalam batch

Perintah berikut melakukan hal yang sama seperti contoh sebelumnya. Satu-satunya perbedaan adalah bahwa hal itu melepaskan instance dalam batch.

Perintah ini hanya akan memodifikasi OpsWorks sumber daya. Status EC2 instance akan tetap sama.

```
python3 layer_detacher.py detach \
--layer-id opsworks-layer-id \
--region opsworks-stack-region \
--batch-size 5
```

Contoh 4: Bersihkan semua sumber daya untuk lapisan dan hapus layer

Perintah berikut akan mengulangi semua sumber daya untuk lapisan dan menghapusnya. Secara lebih rinci, itu akan menghentikan dan menghapus semua instance di OpsWorks dan EC2, melepaskan penyeimbang beban dan membatalkan pendaftaran instans Amazon RDS, elastis dan volume. IPs Setelah membersihkan sumber daya, itu akan menghapus lapisan.

Perintah ini akan menghapus OpsWorks sumber daya dan EC2 instance. Jika Anda ingin EC2 instance Anda tetap tidak tersentuh, gunakan `detach` perintah sebelum menggunakan perintah. `cleanup` Dengan begitu `cleanup` perintah akan menghapus semua sumber daya yang tersisa.

```
python3 layer_detacher.py cleanup \
--layer-id opsworks-layer-id \
--region opsworks-stack-region
```

Contoh 5: Bersihkan semua sumber daya untuk tumpukan dan hapus tumpukan

Perintah berikut akan mengulangi semua lapisan dan kemudian mengulangi sumber daya setiap lapisan. Untuk setiap lapisan, perintah akan menghentikan dan menghapus semua instance di dan EC2, melepaskan penyeimbang beban, OpsWorks dan membatalkan pendaftaran instans Amazon RDS, elastis, dan volume. IPs Kemudian, perintah akan menghapus layer. Proses yang sama akan dilakukan di setiap lapisan yang termasuk dalam tumpukan ini. Akhirnya, setelah semua lapisan dihapus, tumpukan akan dihapus.

Perintah ini akan menghapus OpsWorks sumber daya dan EC2 instance. Jika Anda ingin EC2 instance Anda tetap tidak tersentuh, gunakan `detach` perintah sebelum menggunakan perintah. `cleanup` Dengan begitu `cleanup` perintah akan menghapus semua sumber daya yang tersisa.

```
python3 layer_detacher.py cleanup \
--stack-id opsworks-stack-id \
--region opsworks-stack-region
```

Langkah 4: Lanjutkan mengoperasikan sumber daya Anda setelah melepaskan diri OpsWorks

Setelah menjalankan `detach` perintah, alat membuat AWS Service Catalog AppRegistry aplikasi baru yang sesuai dengan lapisan terpisah. Nama aplikasi mengikuti format*layer-name*`---`*layer-id*. Ini juga menambahkan `OpsWorksLayerId` tag untuk secara unik mengidentifikasi aplikasi yang cocok dengan lapisan terpisah.

Untuk menambahkan AWS sumber daya baru ke aplikasi ini (misalnya, EC2 instance baru), Anda dapat melakukan salah satu hal berikut:

1. Tandai sumber daya dengan tag aplikasi unik AppRegistry aplikasi:

   Kunci tag: `awsApplication`

   Nilai: `arn:aws:resource-groups:`*region*`:`*account-id*`:group/`*application-name*`/`*application-id*`>`
2. Jalankan associate-resourceperintah.

Selain itu, untuk setiap AppRegistry aplikasi, Grup Sumber Daya dibuat. Grup Sumber Daya berisi tag berikut.

Tombol tanda	Nilai
`EnableAWSServiceCatalogAppRegistry`	`TRUE`
`aws:servicecatalog:applicationName`	*`application-name`*
`aws:servicecatalog:applicationId`	*`application-id`*
`aws:servicecatalog:applicationArn`	`arn:aws:servicecatalog:` *`region`*`:`*`account-id`* `:/applications/` *`application-id`*

Melakukan tugas setelah detasemen

Tabel berikut memberikan informasi tentang cara melakukan tugas setelah detasemen:

Tugas	Deskripsi
Menjalankan peristiwa siklus hidup	Setelah menjalankan `detach` perintah dan jika Anda memilih opsi, skrip membuat 5 dokumen Otomasi yang cocok dengan 5 peristiwa OpsWorks siklus hidup. Nama setiap dokumen Otomasi mengikuti format ini:*`layer-id`*`_`*`lifecycle-event`*`_automation_document` . Untuk mensimulasikan OpsWorks perilaku di Systems Manager, Anda harus secara manual memicu eksekusi Otomasi saat menyediakan, menghentikan EC2 instance, atau menerapkan/menghapus resep.

Tugas	Deskripsi
Memperbarui JSON Kustom	JSON khusus untuk tumpukan dan lapisan disimpan dalam bucket S3 yang ditentukan selama detasemen, atau sebagai alternatif di bucket S3 baru yang dibuat. Nama file yang disimpan untuk file JSON adalah sebagai berikut: • `layercustomjson.json` • `stackcustomjson.json`
Mengubah daftar jalankan untuk peristiwa siklus hidup	Daftar jalankan untuk setiap peristiwa siklus hidup ditentukan dalam dokumen Otomasi yang sesuai. Untuk mengubah daftar run, cari dokumen Otomasi dalam AppRegistry aplikasi dan ubah `RunList` parameternya. Proses untuk memperbarui resep dan buku masak tidak berubah karena`AWS-Apply ChefRecipes` , yang dipicu oleh dokumen Otomasi, mendukung sumber yang sama seperti. OpsWorks

Tugas	Deskripsi
Mengelola penyembuhan otomatis/penskalaan otomatis	Saat Anda melepaskan sebuah instance, OpsWorks Agen akan menghapus instalan. Tanpa agen, OpsWorks tidak dapat secara otomatis menyembuhkan atau mengganti instance yang tidak sehat, juga tidak dapat secara otomatis menskalakan armada Anda. Untuk melanjutkan penskalaan otomatis dan mengganti instans yang gagal, buat grup Amazon Auto EC2 Scaling. Grup ini akan meluncurkan instance baru untuk mempertah ankan kapasitas yang diinginkan ketika Amazon EC2 mendeteksi instans tidak sehat yang perlu diganti.
Mengelola Load Balancer	Jika layer Anda menggunakan Classic Load Balancer, `detach` perintah akan melepaska nnya sebelum membatalkan pendaftaran instance. Hal ini dilakukan untuk memastika n bahwa semua asosiasi instans ELB tetap dipertahankan di Amazon EC2 selama proses pelepasan, sehingga mengarah ke nol downtime. Setelah proses selesai, Anda akan dapat mengelola ELB Anda. EC2

Tugas	Deskripsi
Menghubungkan ke instans Anda	Ketika Anda menjalankan `detach` perintah `handle-prerequisites` or, dua pemeriksaan terjadi: • Versi Agen SSM dan izin • Kunci SSH Perintah juga menawarkan opsi untuk memperbarui Agen SSM dan menambahk an izin yang diperlukan sehingga Anda dapat terhubung ke instance menggunakan Session Manager. Jika kunci SSH ada, Anda juga memiliki opsi untuk SSH ke dalam instance.

Menggunakan tab Instans Manajer Aplikasi Systems Manager

Setelah detasemen, Anda akan dapat melihat dan mengelola instance Anda di tab Instans Manajer Aplikasi.

Tab Instances menyediakan informasi agregat tentang EC2 instance aplikasi, seperti status, status kesehatan, dan status perintah terakhir. Dengan menggunakan tab ini, Anda dapat melihat informasi terperinci tentang instance individual seperti riwayat perintah, status alarm, kesehatan agen Systems Manager, dan banyak lagi. Tab Instances juga menyediakan berbagai tindakan, seperti kemampuan untuk menerapkan resep Chef, memulai atau menghentikan instance, atau menambah atau menghapus instance dari grup Auto Scaling.

Memulai dengan AWS OpsWorks Stacks

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks menyediakan serangkaian komponen yang dapat disesuaikan yang dapat Anda campur dan cocokkan untuk membuat tumpukan yang memenuhi tujuan spesifik Anda. Tantangan bagi pengguna baru adalah memahami cara merakit komponen-komponen ini menjadi tumpukan kerja dan mengelolanya secara efektif. Inilah cara Anda bisa memulai.

Jika Anda ingin...	Lengkapi panduan ini:
Buat tumpukan sampel secepat mungkin	Memulai: Contoh
Bereksperimenlah dengan tumpukan berbasis Linux	Memulai: Linux
Bereksperimenlah dengan tumpukan berbasis Windows	Memulai: Windows
Pelajari cara membuat buku masak Chef Anda sendiri	Memulai: Buku Masak

Jika Anda memiliki sumber daya komputasi yang ada—instans Amazon atau bahkan EC2 instans lokal yang berjalan di perangkat keras Anda sendiri—Anda dapat memasukkannya ke dalam tumpukan, bersama dengan instance yang Anda buat dengan Stacks. AWS OpsWorks Anda kemudian dapat menggunakan AWS OpsWorks Stacks untuk mengelola semua instance terkait sebagai grup, terlepas dari bagaimana mereka dibuat.

Support Wilayah

Anda dapat mengakses AWS OpsWorks Stacks secara global; Anda juga dapat membuat dan mengelola instans secara global. Pengguna dapat mengonfigurasi instans AWS OpsWorks Stacks yang akan diluncurkan di AWS wilayah mana pun kecuali AWS GovCloud (AS-Barat) dan Wilayah Tiongkok (Beijing). Untuk bekerja dengan AWS OpsWorks Stacks, instance harus dapat terhubung ke salah satu titik akhir API layanan instans AWS OpsWorks Stacks berikut.

Sumber daya hanya dapat dikelola di wilayah tempat mereka dibuat. Sumber daya yang dibuat di satu titik akhir regional tidak tersedia, juga tidak dapat dikloning ke, titik akhir regional lainnya. Anda dapat meluncurkan instance di salah satu wilayah berikut.

- Wilayah AS Timur (Ohio)
- Wilayah AS Timur (Virginia Utara)
- Wilayah AS Barat (Oregon)
- Wilayah Barat AS (California U.)
- Wilayah Kanada (Tengah) (hanya API, tidak tersedia untuk tumpukan yang dibuat AWS Management Console di.)
- Wilayah Asia Pasifik (Mumbai)
- Wilayah Asia Pasifik (Singapura)
- Wilayah Asia Pasifik (Sydney)
- Wilayah Asia Pacific (Tokyo)
- Wilayah Asia Pasifik (Seoul)
- Wilayah Eropa (Frankfurt)
- Wilayah Eropa (Irlandia)
- Wilayah Eropa (London)
- Wilayah Eropa (Paris)
- Wilayah Amerika Selatan (Sao Paulo)

Memulai dengan Sample Stack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Panduan ini menunjukkan cara menggunakan AWS OpsWorks Stacks untuk membuat contoh lingkungan aplikasi Node.js dengan cepat hanya dengan beberapa klik mouse dan tanpa menulis

kode. Setelah selesai, Anda memiliki instans Amazon Elastic Compute Cloud (Amazon EC2) yang menjalankan Chef 12, server HTTP Node.js, dan aplikasi web yang dapat Anda gunakan untuk berinteraksi dengan Twitter dan meninggalkan komentar di halaman web.

> **Note**
>
> Karena menyelesaikan panduan ini secara otomatis membuat instance dengan tipe c3.large, Anda tidak dapat menggunakan panduan ini, atau alat pembuatan Sample Stack di AWS OpsWorks Stacks, di Tingkat Gratis.AWS Meskipun menggunakan alat pembuatan Sample Stack di VPC membuat instance t2.medium, saat ini tidak VPCs tersedia di Tingkat Gratis.AWS

Langkah 1: Lengkapi Prasyarat

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda harus menyelesaikan langkah-langkah pengaturan berikut sebelum Anda dapat memulai penelusuran. Langkah-langkah penyiapan ini termasuk mendaftar untuk AWS akun, membuat pengguna administratif, dan menetapkan izin akses ke Stacks. AWS OpsWorks

Topik

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif
- Tetapkan Izin Akses Layanan

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimkan email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Tetapkan Izin Akses Layanan

Aktifkan akses ke layanan AWS OpsWorks Stacks (dan layanan terkait yang diandalkan oleh AWS OpsWorks Stacks) dengan menambahkan `AWSOpsWorks_FullAccess` dan `AmazonS3FullAccess` izin ke peran atau pengguna Anda.

Untuk informasi selengkapnya tentang menambahkan izin, lihat Menambahkan izin identitas IAM (konsol).

Anda sekarang telah menyelesaikan semua langkah pengaturan dan dapat memulai penelusuran ini.

Langkah 2: Buat Tumpukan

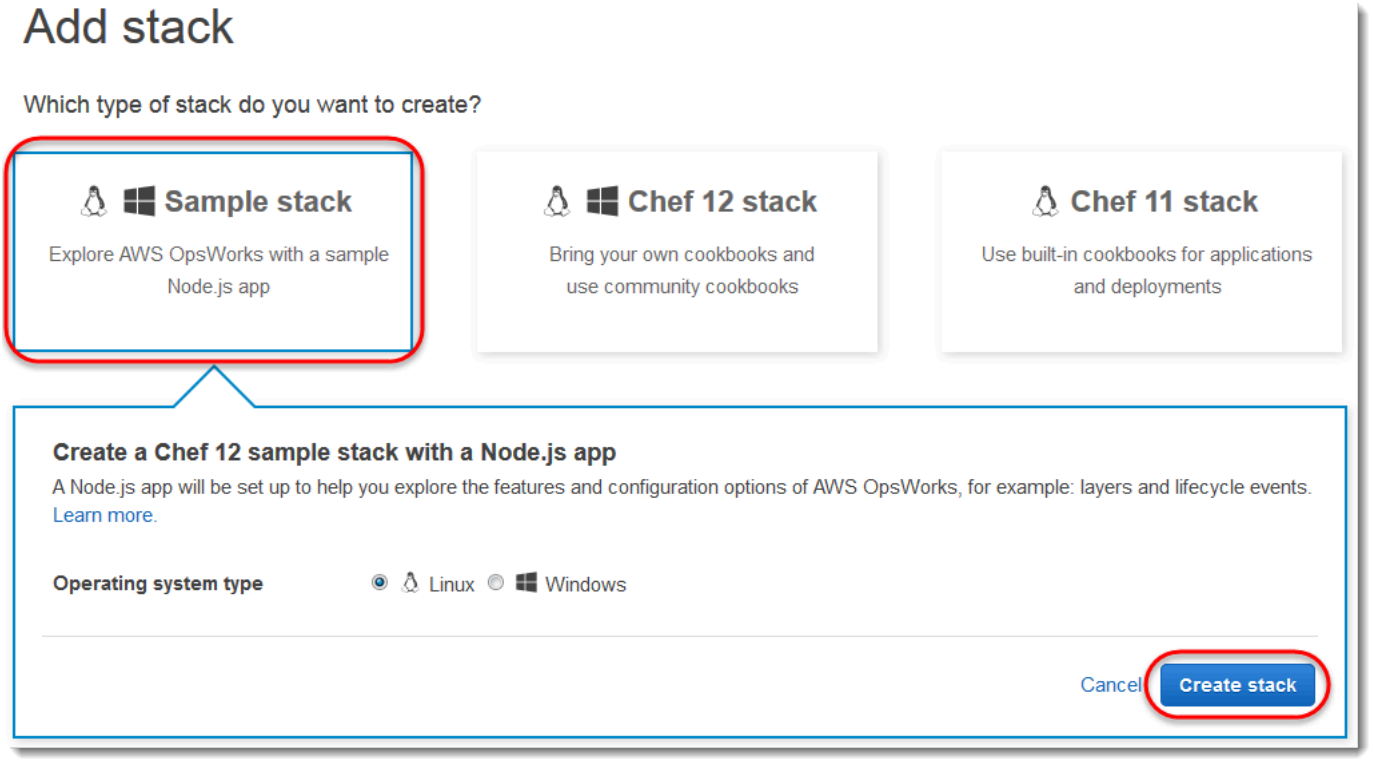
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pada langkah ini, Anda menggunakan konsol AWS OpsWorks Stacks untuk membuat tumpukan. Tumpukan adalah kumpulan instance (seperti instans Amazon EC2) dan AWS sumber daya terkait yang memiliki tujuan bersama dan yang ingin Anda kelola bersama. (Untuk informasi selengkapnya, lihat Tumpukan.) Hanya akan ada satu contoh untuk panduan ini.

Sebelum Anda memulai langkah ini, selesaikan prasyarat.

Untuk membuat tumpukan

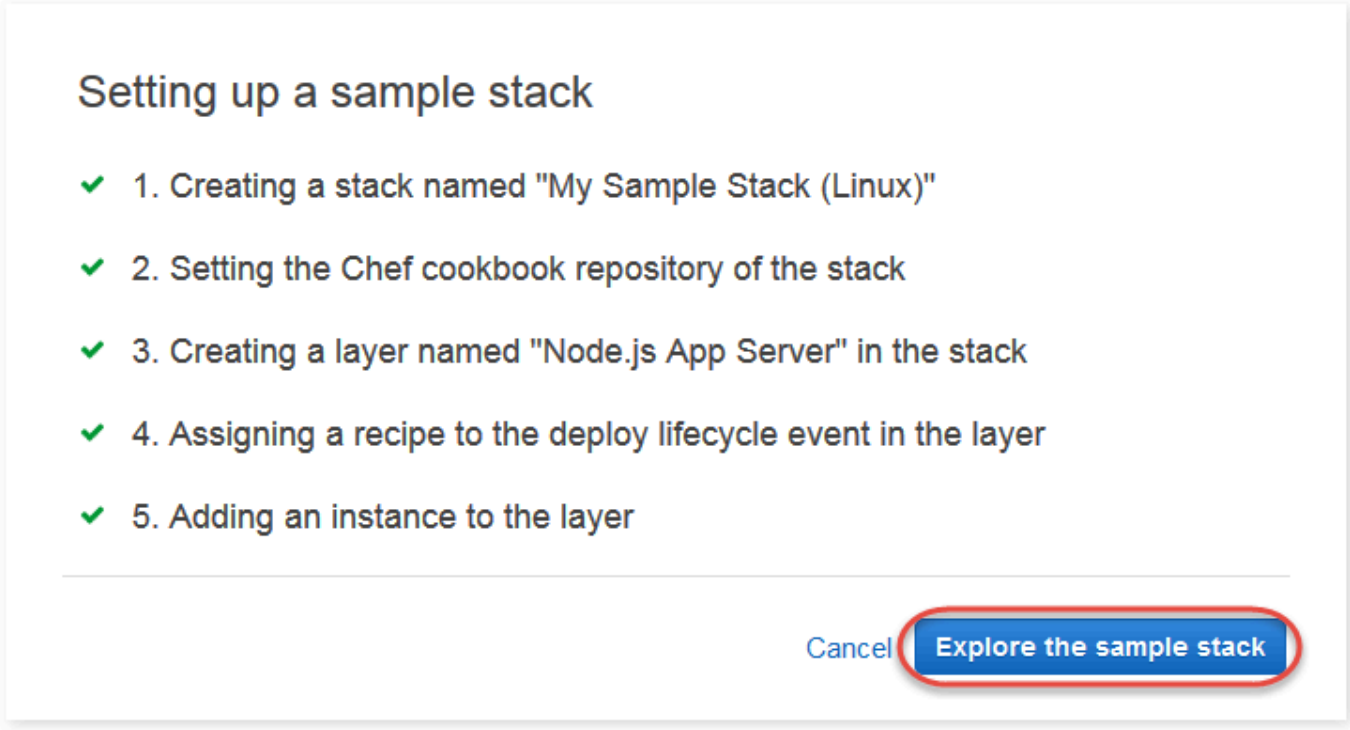
1. Masuk ke AWS Management Console dan buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Lakukan salah satu hal berikut, jika berlaku:
   - Jika halaman Selamat Datang di AWS OpsWorks Tumpukan ditampilkan, pilih Tambahkan tumpukan pertama Anda atau Tambahkan tumpukan AWS OpsWorks tumpukan pertama Anda (kedua pilihan melakukan hal yang sama). Halaman Tambahkan tumpukan ditampilkan.
   - Jika halaman OpsWorks Dasbor ditampilkan, pilih Tambah tumpukan. Halaman Tambahkan tumpukan ditampilkan.
3. Dengan halaman Add stack ditampilkan, pilih Sample stack, jika belum dipilih untuk Anda.
4. Dengan Linux yang sudah dipilih untuk jenis sistem operasi, pilih Buat tumpukan:



5. AWS OpsWorks Stacks membuat tumpukan bernama My Sample Stack (Linux). AWS OpsWorks Stacks juga menambahkan semua komponen yang diperlukan untuk menerapkan aplikasi ke tumpukan:

   - Lapisan, yang merupakan cetak biru untuk satu set instance. Ini menentukan hal-hal seperti pengaturan instance, sumber daya, paket yang diinstal, dan grup keamanan. (Untuk informasi selengkapnya, lihat Lapisan.) Layer ini bernama Node.js App Server.
   - Sebuah contoh, yang dalam hal ini adalah EC2 instance Amazon Linux 2. (Untuk informasi lebih lanjut tentang instance, lihatInstans.) Nama host instance adalah nodejs-server1.
   - Aplikasi, yang merupakan kode untuk dijalankan pada instance. (Untuk informasi selengkapnya tentang aplikasi, lihatAplikasi.) Aplikasi ini bernama Node.js Sample App.

6. Setelah AWS OpsWorks Stacks membuat tumpukan, pilih Jelajahi tumpukan sampel untuk menampilkan halaman My Sample Stack (Linux) (jika Anda menyelesaikan panduan ini beberapa kali, My Sample Stack (Linux) mungkin memiliki nomor berurutan setelahnya, seperti 2 atau 3):



Pada langkah berikutnya, Anda akan memulai instance dan menerapkan aplikasi ke instance.

Langkah 3: Mulai Instans dan Terapkan Aplikasi

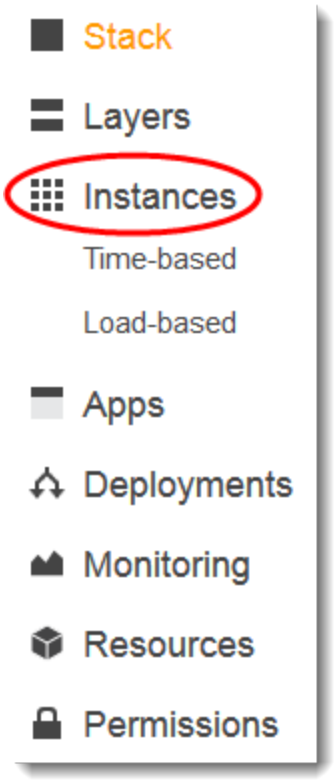
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang setelah Anda memiliki instance dan aplikasi, mulai instance dan terapkan aplikasi ke instance.

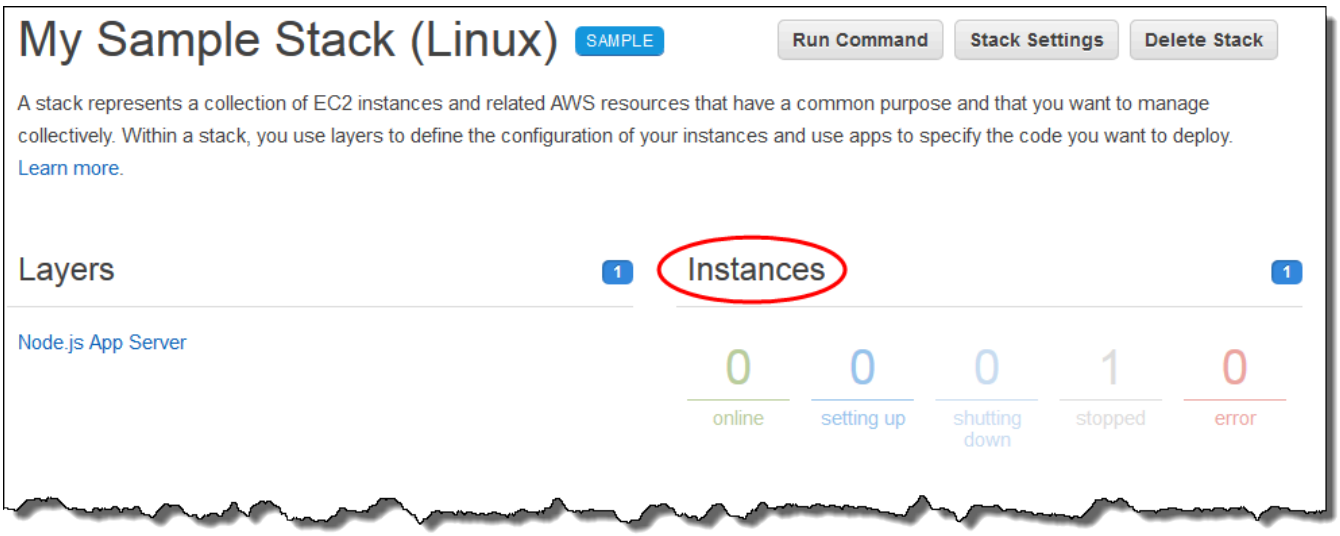
Untuk memulai instance dan menerapkan aplikasi

1. Lakukan salah satu tindakan berikut:

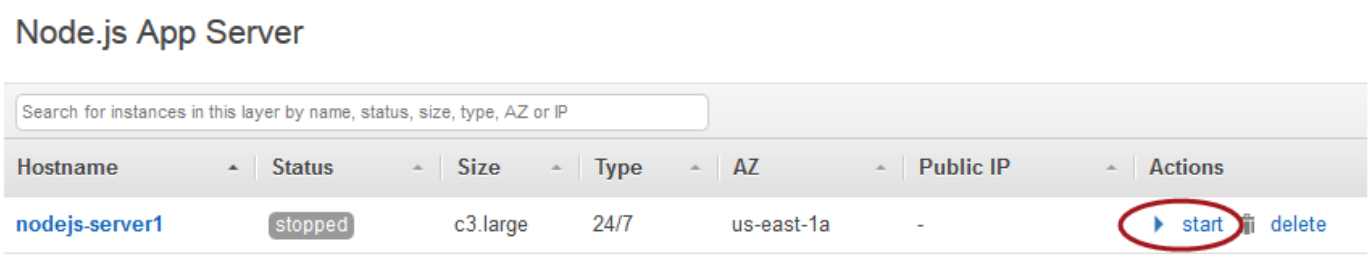
- Di panel navigasi layanan, pilih Instans:



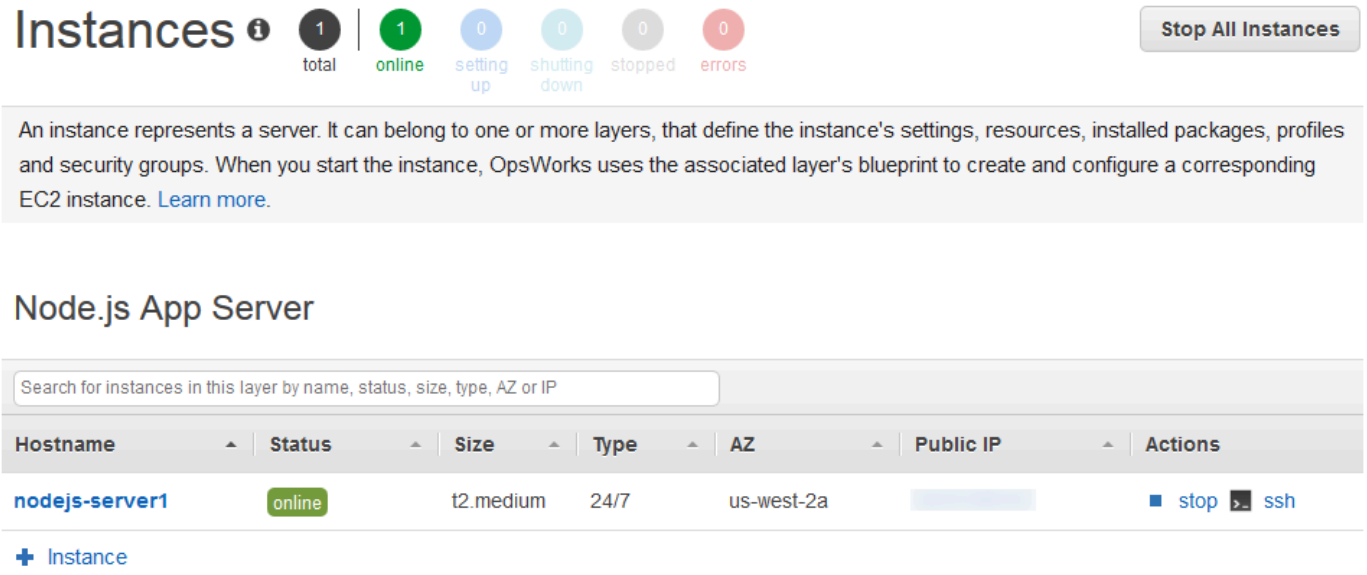
- Pada halaman My Sample Stack (Linux), pilih Instances:



2. Pada halaman Instans, untuk Node.js App Server, untuk nodejs-server1, pilih mulai:



3. Jangan melanjutkan sampai lingkaran online berwarna hijau terang. (Jika Anda melihat pesan kegagalan, konsultasikanPanduan Debugging dan Pemecahan Masalah.)
4. Saat instance sedang disiapkan, AWS OpsWorks Stacks menerapkan aplikasi ke instance.
5. Hasil Anda harus menyerupai tangkapan layar berikut sebelum melanjutkan (jika Anda menerima pesan kegagalan, Anda mungkin ingin berkonsultasi dengan Panduan Debugging dan Pemecahan Masalah.):



Anda sekarang memiliki instance dengan aplikasi yang telah diterapkan ke instance.

Pada langkah berikutnya, Anda menguji aplikasi pada instance.

Langkah 4: Uji Aplikasi yang Diterapkan pada Instans

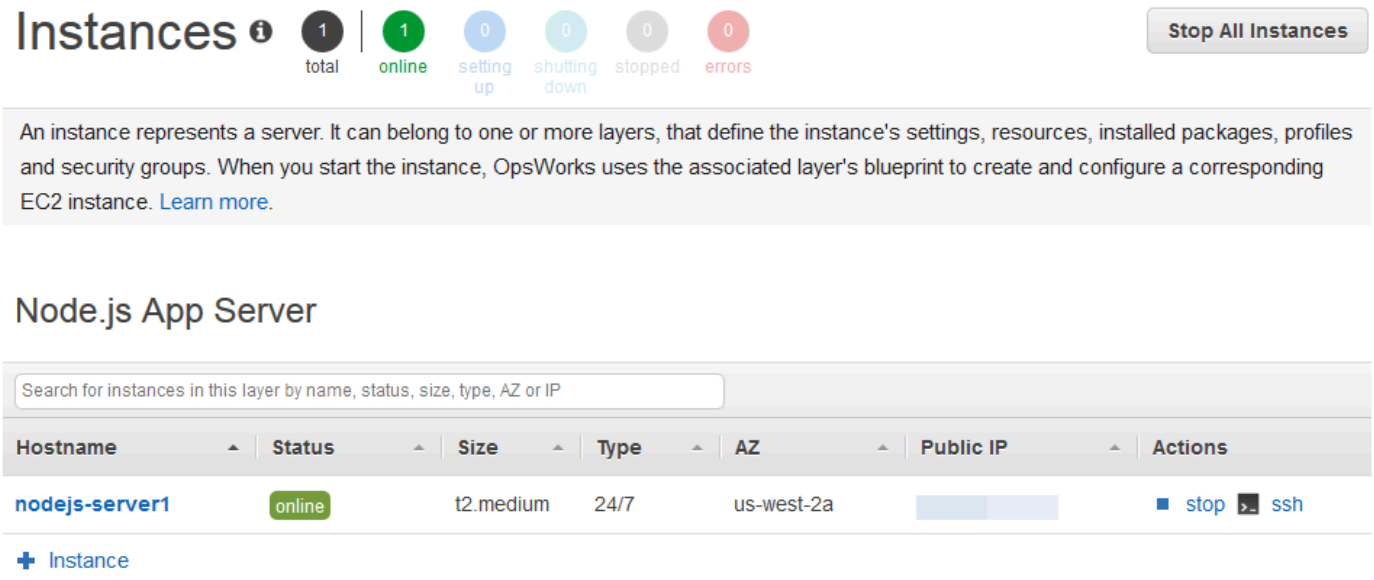
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Uji hasil penerapan aplikasi pada instance.

Untuk menguji penerapan pada instance

1. Dengan halaman Instans ditampilkan dari langkah sebelumnya, untuk Node.js App Server, untuk nodejs-server1, untuk IP Publik, pilih alamat IP.



2. Di halaman web ucapan selamat, di kotak teks Tinggalkan komentar, ketik komentar, lalu pilih Kirim untuk menguji aplikasi. Aplikasi menambahkan komentar Anda ke halaman web. Lanjutkan meninggalkan komentar dan memilih Kirim sesering yang Anda inginkan.



3. Jika Anda memiliki akun Twitter, pilih Tweet atau Ikuti @ AWSOps Works, dan ikuti petunjuk di layar untuk men-tweet tentang aplikasi atau mengikuti @ AWSOps Works.

Anda sekarang telah berhasil menguji aplikasi yang diterapkan pada instance.

Pada langkah-langkah yang tersisa, Anda dapat menggunakan konsol AWS OpsWorks Stacks untuk menjelajahi pengaturan tumpukan dan komponennya. Pada langkah berikutnya, Anda dapat memulai eksplorasi Anda dengan memeriksa pengaturan tumpukan.

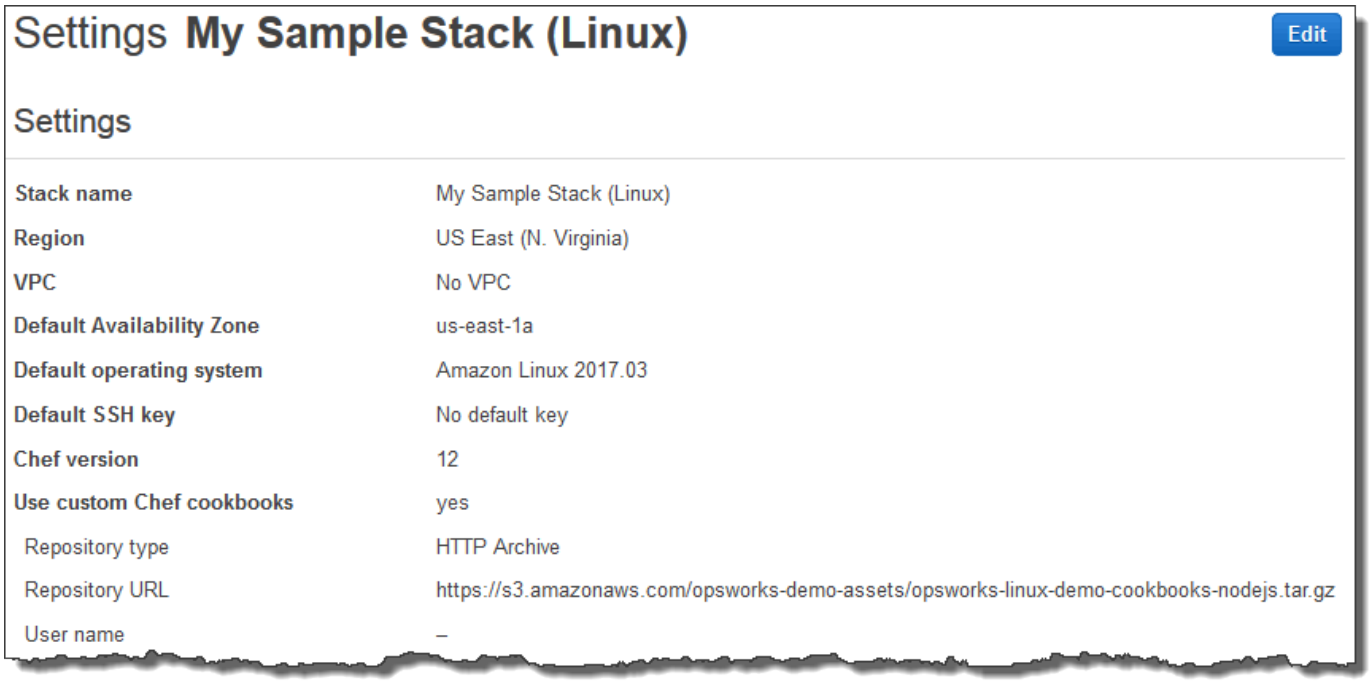
Langkah 5: Jelajahi Pengaturan Stack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Periksa bagaimana AWS OpsWorks Stacks mengatur tumpukan.

Untuk menampilkan pengaturan tumpukan

1. Di bilah navigasi layanan, pilih Stack. Halaman My Sample Stack (Linux) ditampilkan.
2. Pilih Pengaturan Stack. Halaman Pengaturan My Sample Stack (Linux) menampilkan:



Untuk mempelajari lebih lanjut tentang banyak pengaturan, pilih Edit, lalu arahkan kursor ke setiap pengaturan. (Tidak semua pengaturan memiliki deskripsi di layar.) Untuk informasi selengkapnya tentang pengaturan ini, lihat Membuat Tumpukan Baru.

Untuk menjelajahi buku masak Chef yang digunakan dalam panduan ini, buka repositori opsworks-linux-demo-cookbooks-nodejs di. GitHub

Pada langkah berikutnya, Anda dapat menjelajahi pengaturan lapisan.

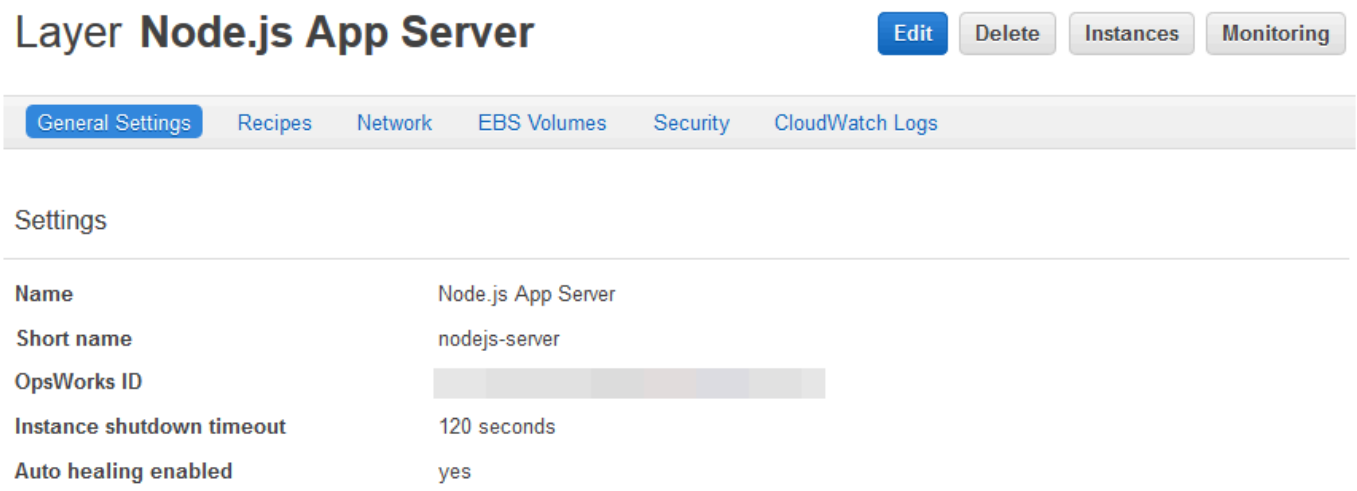
Langkah 6: Jelajahi Pengaturan Layer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Periksa bagaimana AWS OpsWorks Stacks mengatur layer.

Untuk menampilkan pengaturan lapisan

1. Di panel navigasi layanan, pilih Layers. Halaman Layers ditampilkan.
2. Pilih Node.js App Server. Halaman Layer Node.js App Server ditampilkan. Untuk melihat pengaturan lapisan, pilih Pengaturan Umum, Resep, Jaringan, Volume EBS, dan Keamanan:



Untuk mempelajari lebih lanjut tentang banyak pengaturan, pilih Edit, lalu arahkan kursor ke setiap pengaturan. (Tidak semua pengaturan memiliki deskripsi di layar.) Untuk informasi selengkapnya tentang pengaturan ini, lihat Mengedit Konfigurasi OpsWorks Layer.

Pada langkah berikutnya, Anda dapat menjelajahi pengaturan dan log instans.

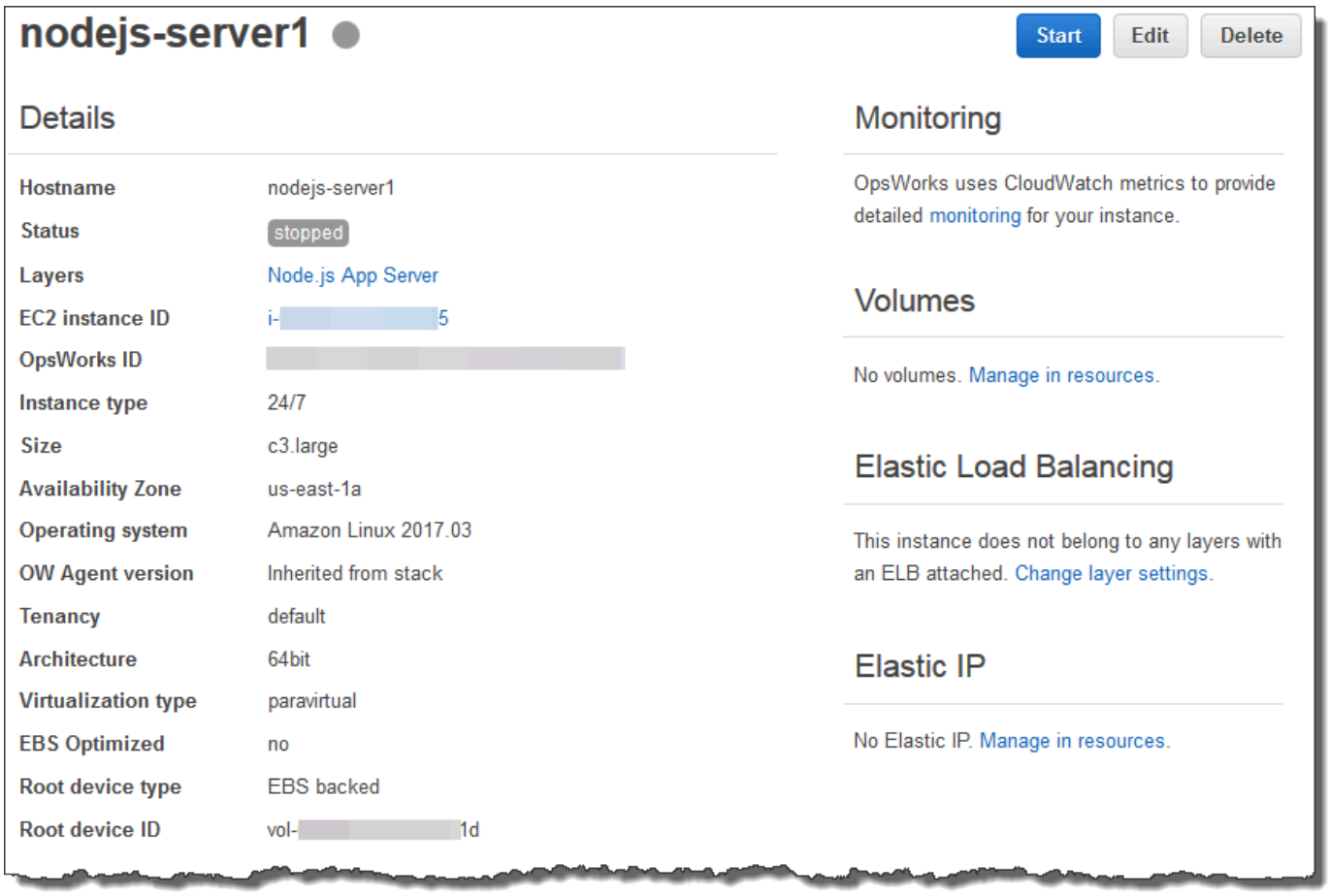
Langkah 7: Jelajahi Pengaturan dan Log Instans

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

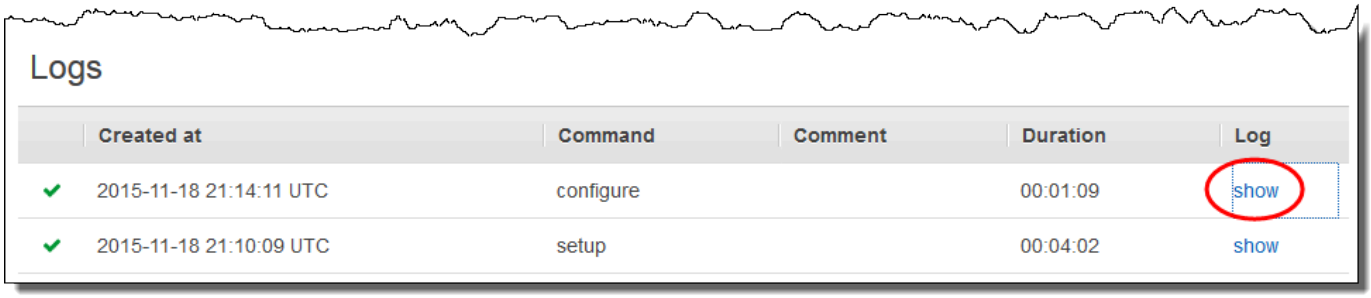
Periksa pengaturan yang digunakan AWS OpsWorks Stacks untuk meluncurkan instance. Anda juga dapat memeriksa log instance yang dibuat oleh AWS OpsWorks Stacks.

Untuk menampilkan pengaturan dan log instans

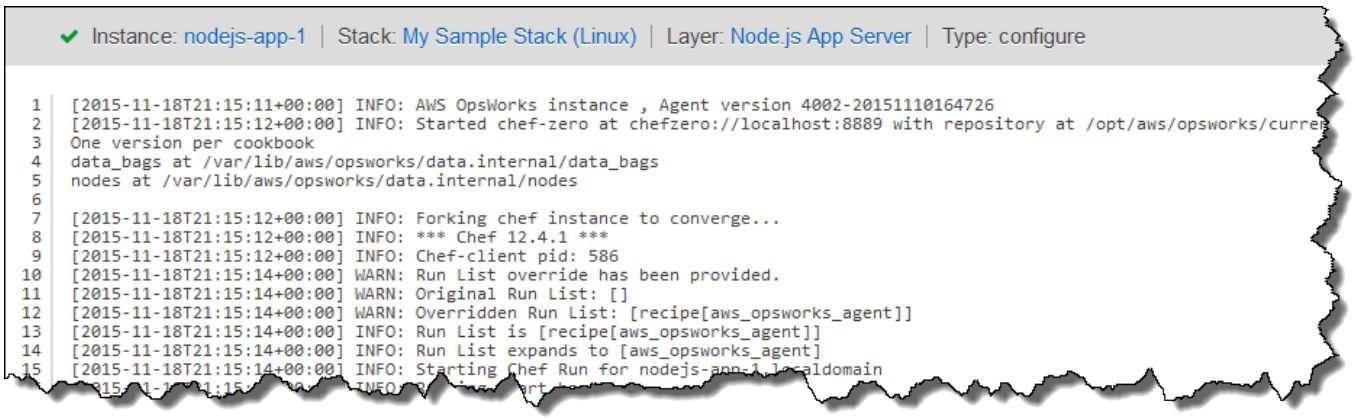
1. Di panel navigasi layanan, pilih Instans. Halaman Instances ditampilkan.
2. Untuk Node.js App Server, pilih nodejs-server1. Halaman properti instance ditampilkan.



3. Untuk menjelajahi log instance, di bagian Log, untuk Log, pilih tampilkan.



4. AWS OpsWorks Tumpukan menampilkan log di tab browser web terpisah.



Untuk mempelajari lebih lanjut tentang apa yang diwakili oleh beberapa pengaturan instans, kembali ke halaman nodejs-server1, pilih Berhenti, dan ketika Anda melihat pesan konfirmasi, pilih Berhenti. Pilih Edit setelah Status berubah dari berhenti menjadi berhenti, lalu arahkan kursor ke setiap pengaturan. (Tidak semua pengaturan memiliki deskripsi di layar.) Untuk informasi selengkapnya tentang pengaturan ini, lihat Menambahkan Instance ke Layer.

Setelah selesai meninjau pengaturan, pilih Mulai untuk memulai ulang instans, dan tunggu hingga Status berubah menjadi online. Jika tidak, Anda tidak akan dapat menguji aplikasi nanti, karena instance akan tetap dihentikan.

> **Note**
>
> Jika Anda ingin masuk ke instance untuk menjelajahinya lebih lanjut, Anda harus terlebih dahulu memberikan AWS OpsWorks Stacks informasi tentang kunci SSH publik Anda (yang dapat Anda buat dengan alat seperti ssh-keygen atau PuTTYgen), dan kemudian Anda harus mengatur izin pada tumpukan My Sample Stack (Linux) untuk memungkinkan pengguna Anda masuk ke instance. Lihat petunjuknya di Mendaftarkan Kunci SSH Publik Pengguna dan Login dengan SSH.

Pada langkah berikutnya, jelajahi pengaturan aplikasi.

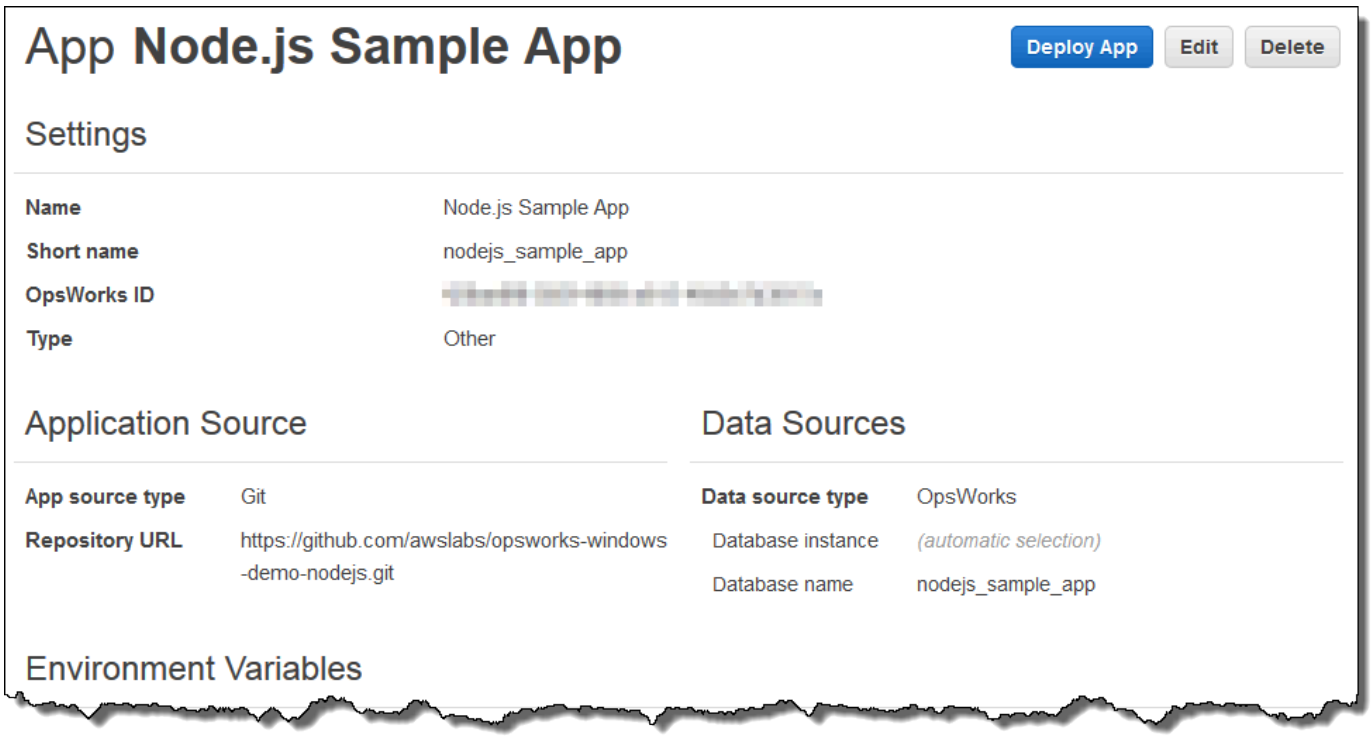
Langkah 8: Jelajahi Pengaturan Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Periksa pengaturan yang digunakan AWS OpsWorks Stacks untuk aplikasi.

Untuk menampilkan pengaturan aplikasi

1. Di panel navigasi layanan, pilih Aplikasi. Halaman Aplikasi ditampilkan.
2. Pilih Aplikasi Sampel Node.js. Halaman Aplikasi Sampel Aplikasi Node.js menampilkan:



Untuk mempelajari tentang apa yang diwakili oleh beberapa pengaturan, pilih Edit, lalu arahkan kursor ke setiap pengaturan. (Tidak semua pengaturan memiliki deskripsi di layar.) Untuk informasi selengkapnya tentang pengaturan ini, lihatMenambahkan Aplikasi.

Pada langkah berikutnya, Anda dapat menjelajahi laporan pemantauan lapisan.

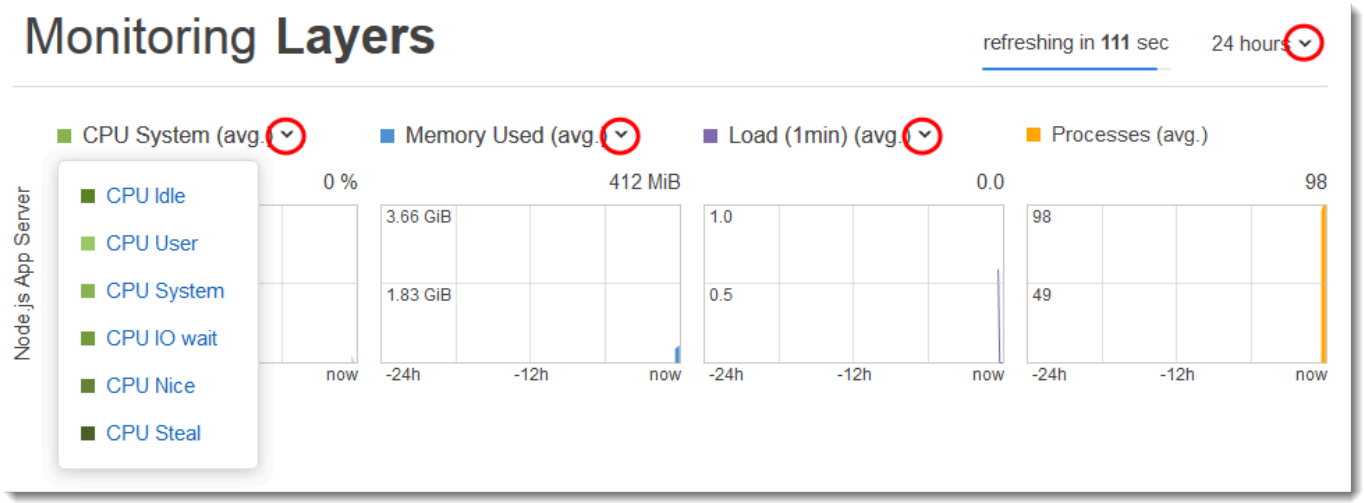
Langkah 9: Jelajahi Laporan Pemantauan Lapisan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Periksa laporan yang dihasilkan AWS OpsWorks Stacks tentang kinerja komputasi lapisan.

Untuk menampilkan laporan pemantauan lapisan

1. Di panel navigasi layanan, pilih Pemantauan. Halaman Monitoring Layers ditampilkan.
2. Untuk menjelajahi tampilan tambahan, pilih panah di samping CPU, Memori, Beban, dan waktu:



Untuk informasi tentang ini dan laporan lainnya, lihat Menggunakan Amazon CloudWatch danPemantauan.

Pada langkah berikutnya, Anda dapat menjelajahi pengaturan tumpukan tambahan.

Langkah 10: Jelajahi Pengaturan Tumpukan Tambahan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pada langkah ini, Anda dapat memeriksa pengaturan tumpukan tambahan.

AWS OpsWorks Stacks tidak menjalankan penerapan terpisah, tidak menyediakan sumber daya tambahan, dan tidak menyesuaikan izin tambahan sebagai bagian dari tumpukan ini, jadi tidak ada banyak minat pada halaman Deployment dan Commands, Resources, dan Izin. Jika Anda ingin melihat pengaturan tersebut, pilih Deployment, Resources, dan Permissions di panel navigasi layanan, masing-masing. Jika Anda ingin informasi lebih lanjut tentang apa yang diwakili oleh halaman-halaman iniMenerapkan Aplikasi, lihatManajemen Sumber Daya, danMengelola izin.

Pada langkah selanjutnya, Anda dapat membersihkan AWS sumber daya yang Anda gunakan untuk panduan ini. Langkah selanjutnya adalah opsional. Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks. Namun, menyimpan AWS sumber daya ini dapat mengakibatkan beberapa tagihan berkelanjutan ke AWS akun Anda. Jika Anda ingin menyimpan AWS sumber daya ini untuk digunakan nanti, Anda sekarang telah menyelesaikan panduan ini, dan Anda dapat melompat ke depan. Langkah Berikutnya

Langkah 11 (Opsional): Bersihkan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk mencegah timbulnya biaya tambahan ke AWS akun Anda, Anda dapat menghapus aplikasi dan AWS sumber daya yang digunakan untuk panduan ini, termasuk instans dan tumpukan Stacks. AWS OpsWorks (Untuk informasi lebih lanjut, lihat AWS OpsWorks Harga.) Namun, Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks. Jika Anda ingin menyimpan AWS sumber daya ini tersedia, Anda sekarang telah menyelesaikan panduan ini, dan Anda dapat melompat ke. Langkah Berikutnya

Konten yang disimpan dalam sumber daya yang Anda buat untuk panduan ini dapat berisi informasi identifikasi pribadi. Jika Anda tidak lagi ingin informasi ini disimpan AWS, ikuti langkah-langkah dalam topik ini.

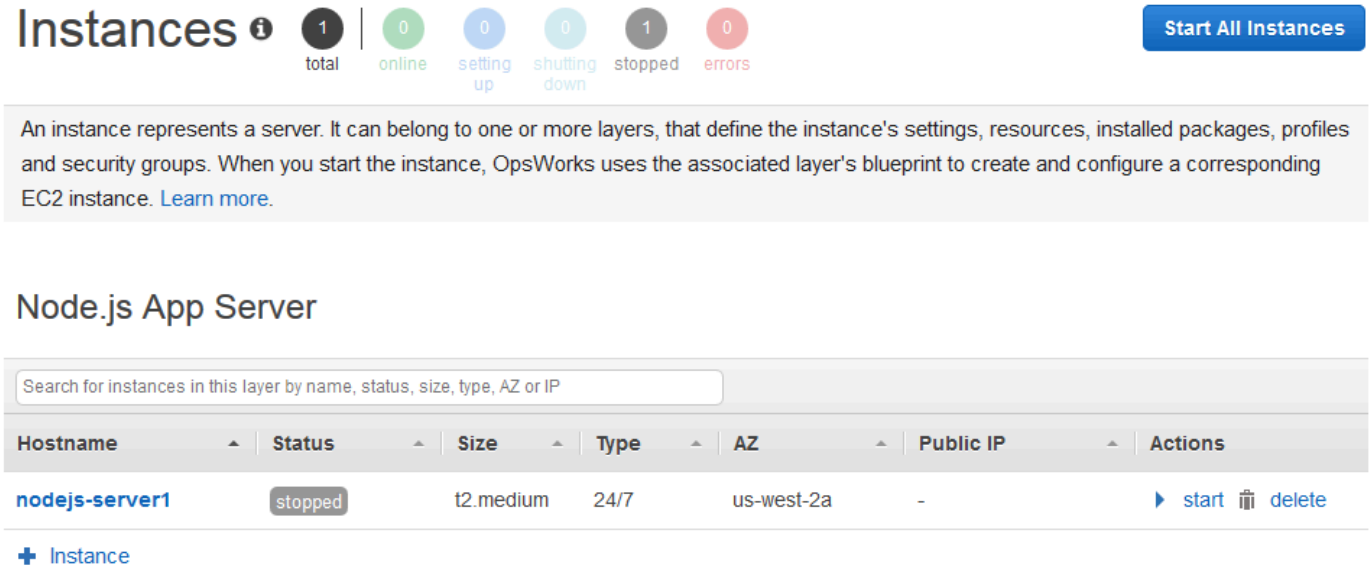
Untuk menghapus aplikasi dari tumpukan

1. Di panel navigasi layanan, pilih Aplikasi. Halaman Aplikasi ditampilkan.
2. Untuk Aplikasi Sampel Node.js, untuk Tindakan, pilih hapus. Saat Anda melihat pesan konfirmasi, pilih Hapus. Saat aplikasi dihapus, Anda melihat pesan Tidak ada aplikasi.

Untuk menghapus instance untuk tumpukan

1. Di panel navigasi layanan, pilih Instans. Halaman Instances ditampilkan.
2. Untuk Node.js App Server, untuk nodejs-server1, untuk Actions, pilih stop. Saat Anda melihat pesan konfirmasi, pilih Berhenti.

   Proses ini dapat menghabiskan waktu beberapa menit. Ketika AWS OpsWorks Stacks selesai, hasil berikut akan ditampilkan.



3. Untuk Tindakan, pilih hapus. Saat Anda melihat pesan konfirmasi, pilih Hapus. Instance dihapus, dan pesan No instance ditampilkan.

Untuk menghapus tumpukan

1. Di panel navigasi layanan, pilih Stack. Halaman My Sample Stack (Linux) ditampilkan.
2. Pilih Hapus Tumpukan. Saat Anda melihat pesan konfirmasi, pilih Hapus. Tumpukan dihapus, dan halaman OpsWorksDasbor ditampilkan.

Secara opsional, Anda dapat menghapus user dan Amazon EC2 key pair yang Anda gunakan untuk panduan ini, jika Anda tidak ingin menggunakannya kembali untuk akses ke layanan dan instance lain AWS . EC2 Untuk petunjuk, lihat Menghapus pengguna IAM dan pasangan EC2 kunci Amazon dan instance Linux.

Anda sekarang telah menyelesaikan panduan ini. Untuk informasi selengkapnya, lihat Langkah Berikutnya.

Langkah Berikutnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang setelah Anda menyelesaikan panduan ini, Anda dapat mempelajari lebih lanjut tentang menggunakan AWS OpsWorks Stacks:

- Berlatih membuat ulang tumpukan ini secara manual sendiri menggunakan AWS OpsWorks Stacks. Lihat Memulai: Linux.
- Jelajahi buku masak dan aplikasi yang digunakan AWS OpsWorks Stacks untuk panduan ini. Lihat Pelajari Lebih Lanjut: Jelajahi Buku Masak yang Digunakan dalam Panduan Ini dan Pelajari Lebih Lanjut: Jelajahi Aplikasi yang Digunakan dalam Panduan Ini dalam Memulai: Linux panduan pendamping.
- Berlatih menggunakan AWS OpsWorks Stacks dengan instance Windows. Lihat Memulai: Windows.
- Pelajari lebih lanjut tentang tumpukan dengan mempelajari caranyaMembuat Tumpukan Baru.
- Pelajari lebih lanjut tentang layer byMengedit Konfigurasi OpsWorks Layer.
- Pelajari lebih lanjut tentang instans olehMenambahkan Instance ke Layer.
- Pelajari lebih lanjut tentang aplikasi olehMenerapkan Aplikasi.
- Pelajari lebih lanjut tentang Buku Masak dan Resep.
- Buat buku masak Anda sendiri. Lihat Memulai: Buku Masak.
- Pelajari tentang mengontrol akses ke tumpukan dengan Keamanan dan Izin.

Memulai dengan Linux Stacks

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Dalam panduan ini, Anda akan belajar cara menggunakan AWS OpsWorks Stacks untuk membuat lingkungan aplikasi Node.js. Setelah selesai, Anda akan memiliki instans Amazon Elastic Compute Cloud (Amazon EC2) yang menjalankan Chef 12, server HTTP Node.js, dan aplikasi web yang dapat Anda gunakan untuk berinteraksi dengan Twitter dan meninggalkan komentar di halaman web.

Chef adalah kerangka kerja pihak ketiga untuk mengonfigurasi dan memelihara server, seperti EC2 instance, dan bagaimana aplikasi digunakan dan dipelihara di server tersebut. Jika Anda tidak terbiasa dengan Chef, setelah menyelesaikan panduan ini, kami sarankan Anda mempelajari lebih lanjut tentang Chef sehingga Anda dapat memanfaatkan sepenuhnya semua yang ditawarkan AWS OpsWorks Stacks. (Untuk informasi selengkapnya, lihat situs web Learn Chef.)

AWS OpsWorks Stacks mendukung empat distribusi Linux: Amazon Linux, Ubuntu Server, CentOS, dan Red Hat Enterprise Linux. Untuk panduan ini, kami menggunakan Ubuntu Server. AWS OpsWorks Stacks juga berfungsi dengan Windows Server. Meskipun kami memiliki panduan yang setara untuk tumpukan Windows Server, kami sarankan Anda menyelesaikan panduan ini terlebih dahulu untuk mempelajari konsep dasar tentang AWS OpsWorks Stacks dan Chef yang tidak diulang di sana. Setelah Anda menyelesaikan panduan ini, lihat langkah-langkahnya. Memulai: Windows

Topik

- Langkah 1: Lengkapi Prasyarat
- Langkah 2: Buat Tumpukan
- Langkah 3: Tambahkan Layer ke Stack
- Langkah 4: Tentukan Aplikasi yang akan Diterapkan ke Instans
- Langkah 5: Luncurkan Instance
- Langkah 6: Menerapkan Aplikasi ke Instans
- Langkah 7: Uji Aplikasi yang Diterapkan pada Instans
- Langkah 8 (Opsional): Bersihkan
- Langkah Berikutnya
- Pelajari Lebih Lanjut: Jelajahi Buku Masak yang Digunakan dalam Panduan Ini
- Pelajari Lebih Lanjut: Jelajahi Aplikasi yang Digunakan dalam Panduan Ini

Langkah 1: Lengkapi Prasyarat

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Selesaikan langkah-langkah pengaturan berikut sebelum Anda dapat memulai penelusuran. Langkah-langkah penyiapan ini termasuk mendaftar untuk AWS akun, membuat pengguna administratif, dan menetapkan izin akses ke Stacks. AWS OpsWorks

Jika Anda telah menyelesaikan Panduan Memulai: Contoh, maka Anda telah memenuhi prasyarat untuk panduan ini, dan Anda dapat melompat ke depan. Langkah 2: Buat Tumpukan

Topik

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif
- Tetapkan Izin Akses Layanan

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai

praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.
2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Tetapkan Izin Akses Layanan

Aktifkan akses ke layanan AWS OpsWorks Stacks (dan layanan terkait yang diandalkan oleh AWS OpsWorks Stacks) dengan menambahkan `AWSOpsWorks_FullAccess` dan `AmazonS3FullAccess` izin ke peran atau pengguna Anda.

Untuk informasi selengkapnya tentang menambahkan izin, lihat Menambahkan izin identitas IAM (konsol).

Anda sekarang telah menyelesaikan semua langkah pengaturan dan dapat memulai panduan ini.

Langkah 2: Buat Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda akan menggunakan konsol AWS OpsWorks Stacks untuk membuat tumpukan. Tumpukan adalah kumpulan contoh dan AWS sumber daya terkait yang memiliki tujuan bersama dan yang ingin Anda kelola bersama. (Untuk informasi selengkapnya, lihat Tumpukan.) Untuk panduan ini, hanya ada satu contoh.

Sebelum Anda mulai, lengkapi prasyarat jika Anda belum melakukannya.

Untuk membuat tumpukan

1. Masuk ke AWS Management Console dan buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Lakukan salah satu hal berikut, jika berlaku:
   - Jika halaman Selamat Datang di AWS OpsWorks Tumpukan ditampilkan, pilih Tambahkan tumpukan pertama Anda atau Tambahkan tumpukan AWS OpsWorks tumpukan pertama Anda (kedua pilihan melakukan hal yang sama). Halaman Tambahkan tumpukan ditampilkan.
   - Jika halaman OpsWorks Dasbor ditampilkan, pilih Tambah tumpukan. Halaman Tambahkan tumpukan ditampilkan.
3. Dengan halaman Tambahkan tumpukan ditampilkan, pilih tumpukan Chef 12 jika belum dipilih untuk Anda.
4. Di kotak nama Stack, ketikkan nama, misalnya**`MyLinuxDemoStack`**. (Anda dapat mengetikkan nama yang berbeda, tetapi pastikan untuk menggantinya `MyLinuxDemoStack` di seluruh panduan ini.)
5. Untuk Wilayah, pilih AS Barat (Oregon).
6. Untuk VPC, lakukan salah satu hal berikut:
   - Jika VPC tersedia, pilihlah. (Untuk informasi selengkapnya, lihat Menjalankan Stack di VPC.)
   - Jika tidak, pilih Tanpa VPC.
7. Untuk sistem operasi Default, pilih Linux dan Ubuntu 18.04 LTS.
8. Untuk Gunakan buku masak Chef khusus, pilih Ya.
9. Untuk jenis Repositori, pilih Http Archive.
10. Untuk URL Repositori, ketik **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`**
11. Biarkan default sebagai berikut:
    - Zona Ketersediaan Default (us-west-2a)

- Kunci SSH default (Jangan gunakan kunci SSH default)
- Nama pengguna (kosong)
- Kata sandi (kosong)
- Warna tumpukan (biru tua)

12. Pilih Lanjutan.
13. Untuk peran IAM, lakukan salah satu hal berikut (untuk informasi lebih lanjut, lihatMengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda):

    - Jika aws-opsworks-service-roletersedia, pilihlah.
    - Jika tidak aws-opsworks-service-roletersedia, pilih Peran IAM Baru.

14. Untuk profil instans IAM Default, lakukan salah satu hal berikut (untuk informasi selengkapnya, lihatMenentukan Izin untuk Aplikasi yang Berjalan pada instance EC2):

    - Jika aws-opsworks-ec2 peran tersedia, pilihlah.
    - Jika aws-opsworks-ec2 peran tidak tersedia, pilih Profil instans IAM baru.

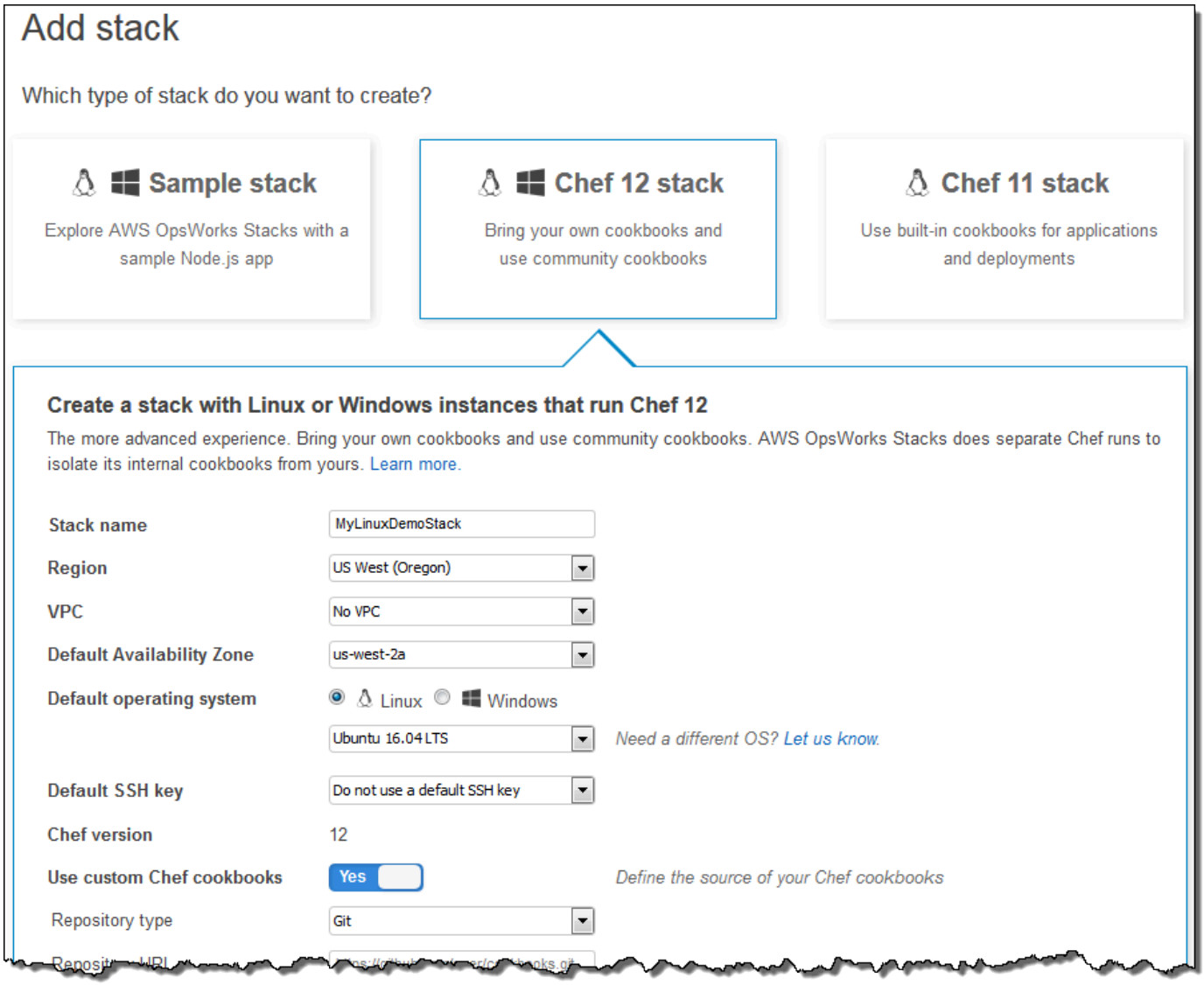
15. Untuk wilayah titik akhir API, pilih titik akhir API regional yang Anda inginkan terkait dengan tumpukan. Jika Anda ingin tumpukan berada di Wilayah AS Barat (Oregon) di titik akhir regional AS Timur (Virginia N.), pilih us-east-1. Jika Anda ingin tumpukan berada di Wilayah Barat AS (Oregon) dan terkait dengan titik akhir regional AS Barat (Oregon), pilih us-west-2.

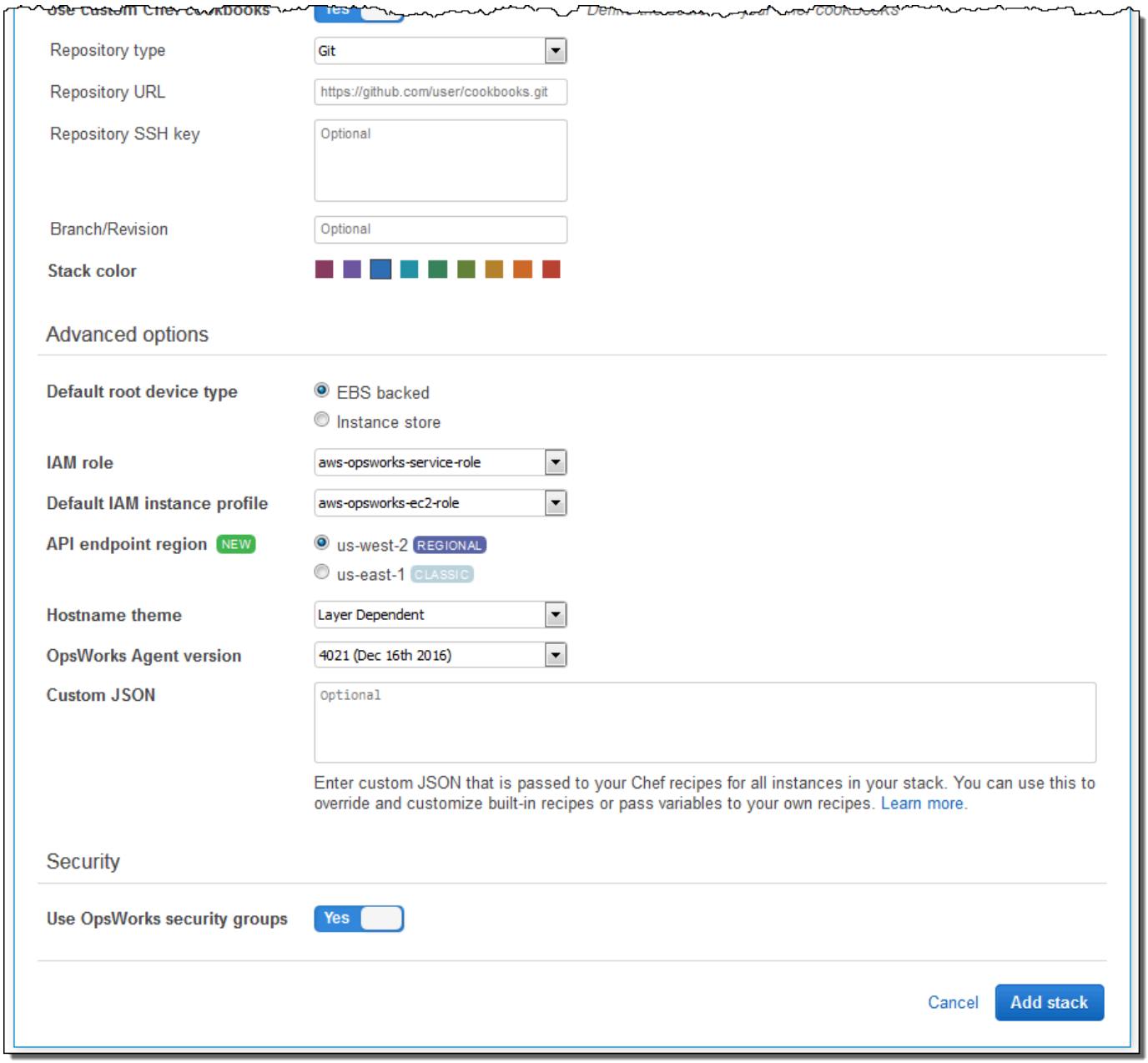
    > **Note**
    >
    > Titik akhir regional AS Timur (Virginia N.) mencakup yang lebih lama Wilayah AWS untuk kompatibilitas mundur, tetapi merupakan praktik terbaik untuk memilih titik akhir regional yang paling dekat dengan tempat Anda mengelola. AWS Untuk informasi selengkapnya, lihat Support Wilayah.

16. Biarkan default sebagai berikut:

    - Jenis perangkat root default (didukung EBS)
    - Tema nama host (Layer Dependent)
    - OpsWorks Versi agen (versi terbaru)
    - Kustom JSON (kosong)
    - Gunakan grup OpsWorks keamanan (Ya)

17. Hasil Anda harus cocok dengan tangkapan layar berikut kecuali mungkin VPC, peran IAM, dan profil instance IAM Default:





18. Pilih Tambahkan Tumpukan. AWS OpsWorks Stacks membuat tumpukan dan menampilkan MyLinuxDemoStackhalaman.

Anda sekarang memiliki tumpukan dengan pengaturan yang benar untuk panduan ini.

Pada langkah berikutnya, Anda akan menambahkan lapisan ke tumpukan.

Langkah 3: Tambahkan Layer ke Stack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

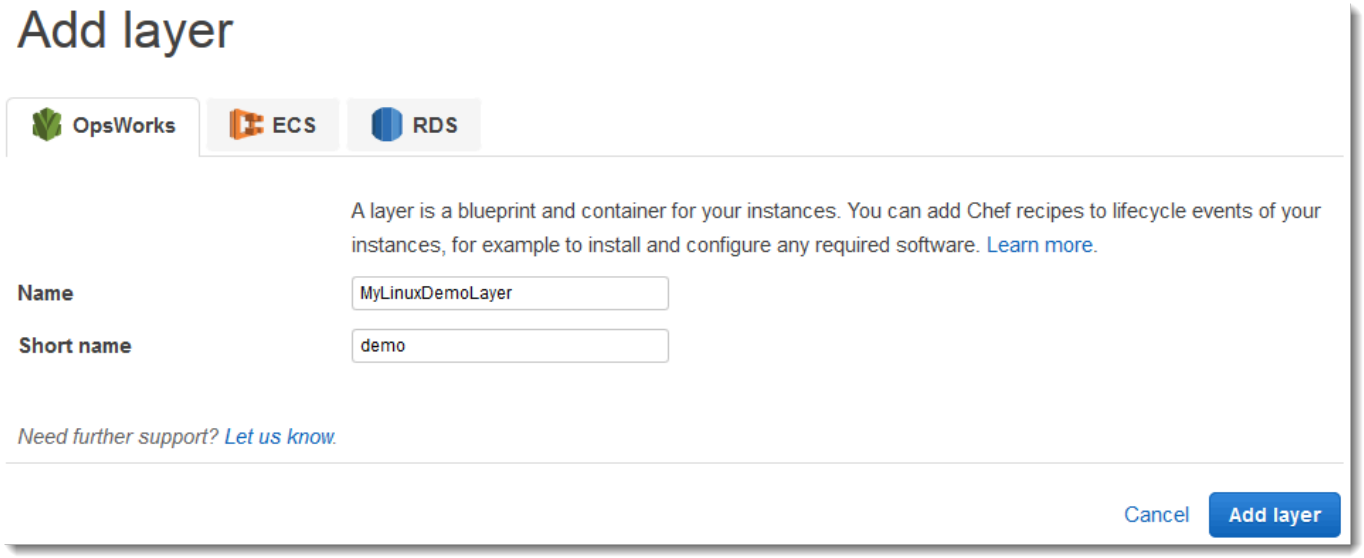
Lapisan adalah cetak biru untuk satu set instance, seperti instans Amazon. EC2 Ini menentukan informasi seperti pengaturan instans, sumber daya, paket yang diinstal, dan kelompok keamanan. Selanjutnya, tambahkan layer ke stack. (Untuk informasi lebih lanjut tentang lapisan, lihatLapisan.)

Untuk menambahkan lapisan ke tumpukan

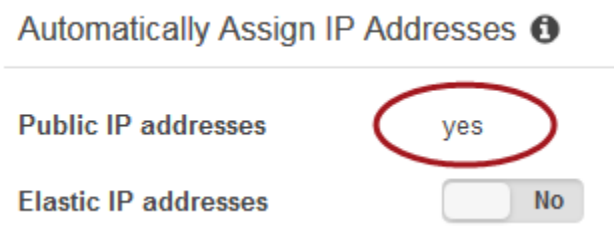
1. Dengan MyLinuxDemoStackhalaman yang ditampilkan dari langkah sebelumnya, untuk Layers, pilih Add a layer:



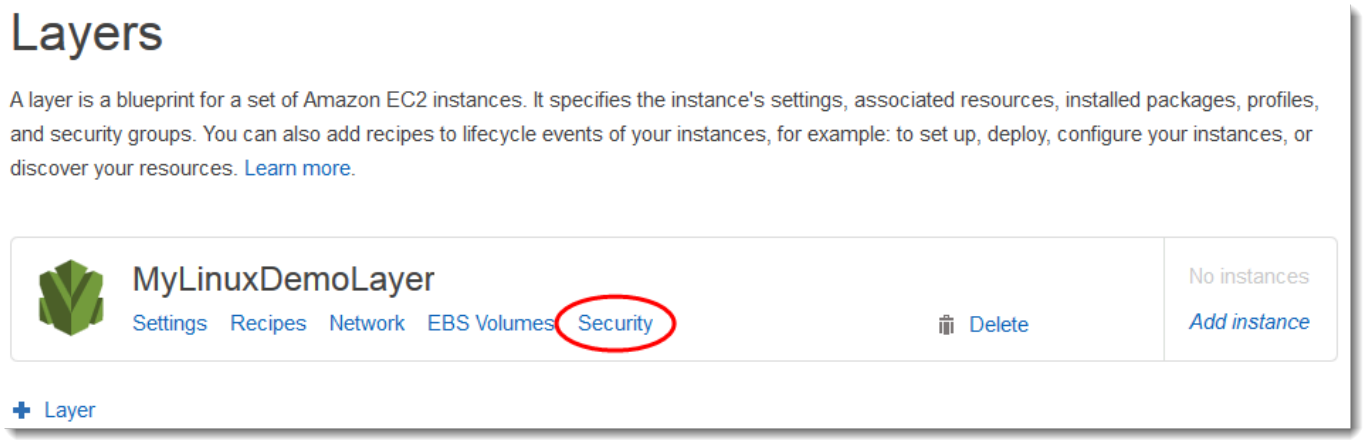
2. Halaman Add Layer ditampilkan. Pada OpsWorkstab, untuk Nama, ketik**`MyLinuxDemoLayer`**. (Anda dapat mengetikkan nama yang berbeda, tetapi pastikan untuk menggantinya `MyLinuxDemoLayer` di seluruh panduan ini.)
3. Untuk Nama pendek, ketik **`demo`** (Anda dapat mengetikkan nilai yang berbeda, tetapi pastikan untuk menggantinya `demo` di seluruh panduan ini):



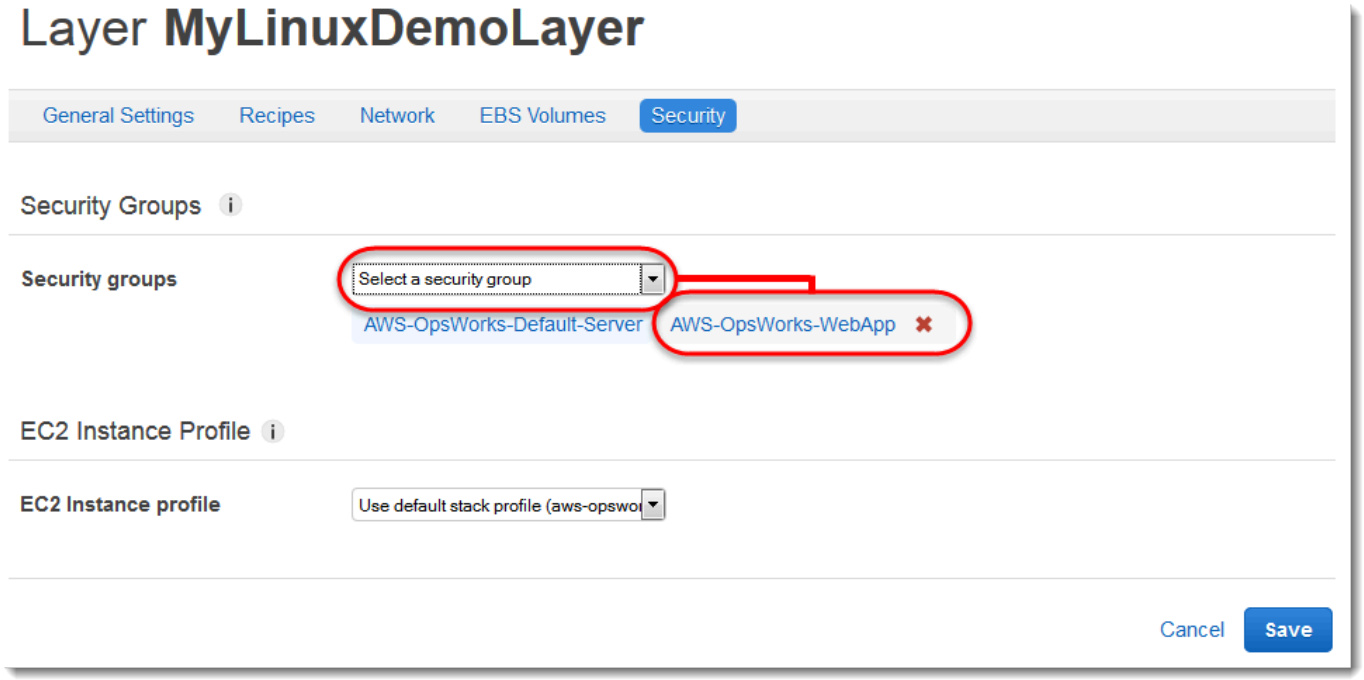
4. Pilih Tambahkan lapisan. AWS OpsWorks Stacks membuat layer dan menampilkan halaman Layers.
5. Pada halaman Layers, untuk MyLinuxDemoLayer, pilih Network.
6. Pada tab Jaringan, di bawah Secara Otomatis Menetapkan Alamat IP, verifikasi bahwa alamat IP Publik diatur ke ya. Jika Anda telah membuat perubahan, pilih Simpan.



7. Pada halaman Layers, pilih Security:



8. MyLinuxDemoLayerHalaman Layer ditampilkan dengan tab Security terbuka. Untuk grup Keamanan, pilih AWS- OpsWorks - WebApp, lalu pilih Simpan:



9. Grup `AWS-OpsWorks-WebApp` keamanan ditambahkan ke lapisan. (Grup keamanan ini memungkinkan pengguna untuk terhubung ke aplikasi pada instance nanti dalam panduan ini. Tanpa grup keamanan ini, pengguna akan menerima pesan di browser web mereka bahwa mereka tidak dapat terhubung ke instance.)

Anda sekarang memiliki lapisan dengan pengaturan yang benar untuk panduan ini.

Pada langkah berikutnya, Anda akan menentukan aplikasi yang akan diterapkan ke instance.

Langkah 4: Tentukan Aplikasi yang akan Diterapkan ke Instans

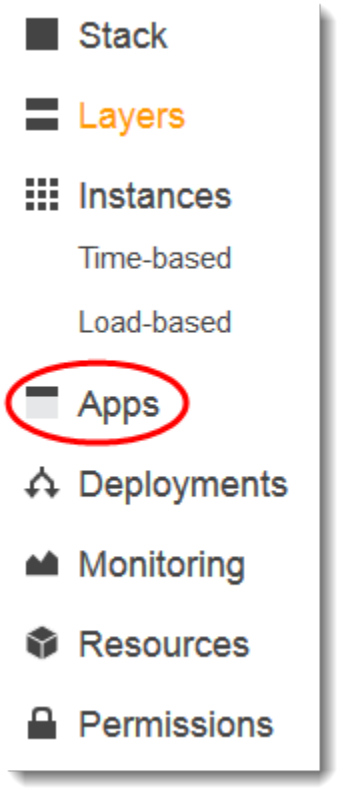
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Beri tahu AWS OpsWorks Stacks tentang aplikasi yang akan Anda terapkan ke instance nanti dalam panduan ini. Dalam konteks ini, AWS OpsWorks Stacks mendefinisikan aplikasi sebagai kode yang ingin Anda jalankan pada sebuah instance. (Untuk informasi selengkapnya, lihat Aplikasi.)

Prosedur di bagian ini berlaku untuk Chef 12 dan tumpukan yang lebih baru. Untuk informasi tentang cara menambahkan aplikasi ke layer di tumpukan Chef 11, lihatLangkah 2.4: Membuat dan Menyebarkan Aplikasi - Chef 11.

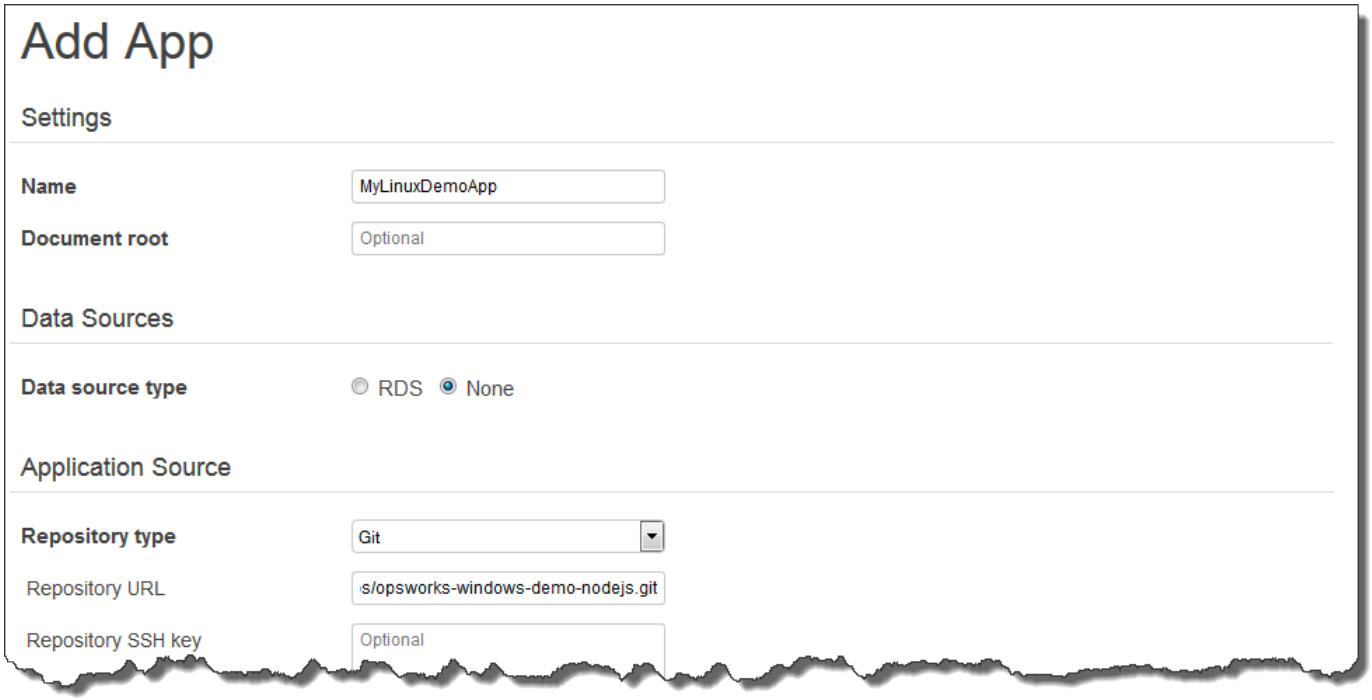
Untuk menentukan aplikasi yang akan diterapkan

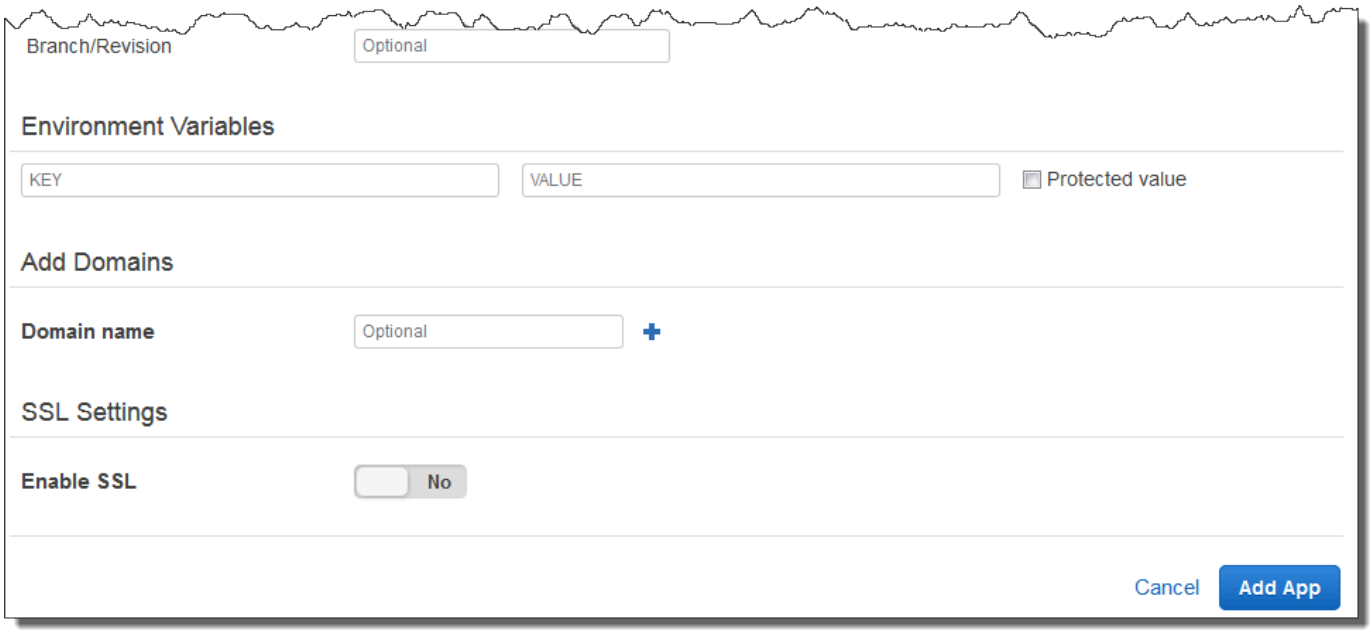
1. Di panel navigasi layanan, pilih Aplikasi:



2. Halaman Aplikasi ditampilkan. Pilih Tambahkan aplikasi. Halaman Tambah Aplikasi ditampilkan.
3. Untuk Pengaturan, untuk Nama, ketik**`MyLinuxDemoApp`**. (Anda dapat mengetikkan nama yang berbeda, tetapi pastikan untuk menggantinya `MyLinuxDemoApp` di seluruh panduan ini.)
4. Untuk Sumber Aplikasi, untuk URL Repositori, ketik **`https://github.com/awslabs/opsworks-windows-demo-nodejs.git`**
5. Biarkan default sebagai berikut:

- Pengaturan, Root dokumen (kosong)
- Sumber Data, Tipe sumber data (Tidak ada)
- Jenis repositori (Git)
- Kunci SSH repositori (kosong)
- Cabang/Revisi (kosong)
- Variabel Lingkungan (KUNCI kosong, NILAI kosong, Nilai Dilindungi yang tidak dicentang)
- Tambahkan Domain, Nama Domain (kosong)
- Pengaturan SSL, Aktifkan SSL (Tidak)





6. Pilih Tambahkan Aplikasi. AWS OpsWorks Stacks menambahkan aplikasi dan menampilkan halaman Aplikasi.

Anda sekarang memiliki aplikasi dengan pengaturan yang benar untuk panduan ini.

Pada langkah berikutnya, Anda akan meluncurkan instance.

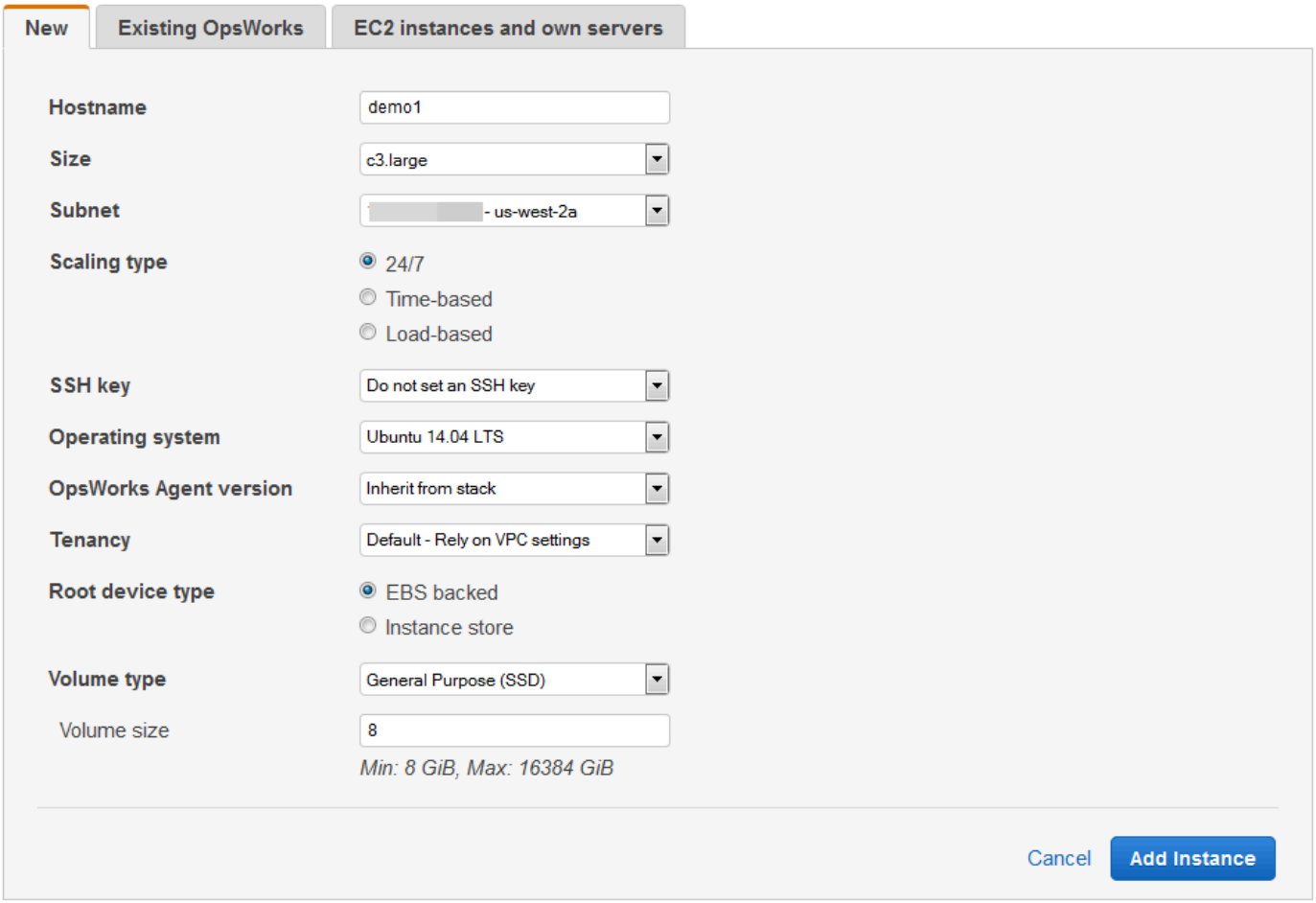
Langkah 5: Luncurkan Instance

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

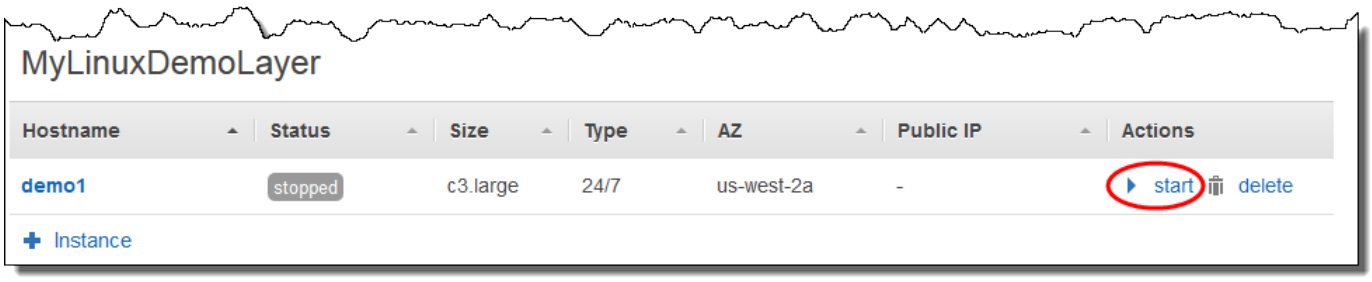
Gunakan AWS OpsWorks Stacks untuk memulai EC2 instance Amazon Server Ubuntu. Instance ini menggunakan pengaturan yang Anda tentukan di lapisan yang Anda buat sebelumnya dalam panduan ini. (Untuk informasi selengkapnya, lihat Instans.)

Untuk meluncurkan instance

1. Di panel navigasi layanan, pilih Instans. Halaman Instans ditampilkan.
2. Untuk MyLinuxDemoLayer, pilih Tambahkan instance.
3. Pada tab Baru, biarkan default untuk yang berikut ini:
   - Nama host (demo1)
   - Ukuran (c3.large)
   - Subnet (*IP address*us-barat-2a)
4. Pilih Lanjutan.
5. Biarkan default sebagai berikut:
   - Jenis penskalaan (24/7)
   - Kunci SSH (Jangan gunakan kunci SSH default)
   - Sistem operasi (Ubuntu 18.04 LTS)
   - OpsWorks Versi agen (Mewarisi dari tumpukan)
   - Penyewaan (Default - Mengandalkan pengaturan VPC)
   - Jenis perangkat root (didukung EBS)
   - Jenis volume (Tujuan Umum (SSD))
   - Ukuran volume (8)
6. Hasil Anda akan mirip dengan tangkapan layar berikut:



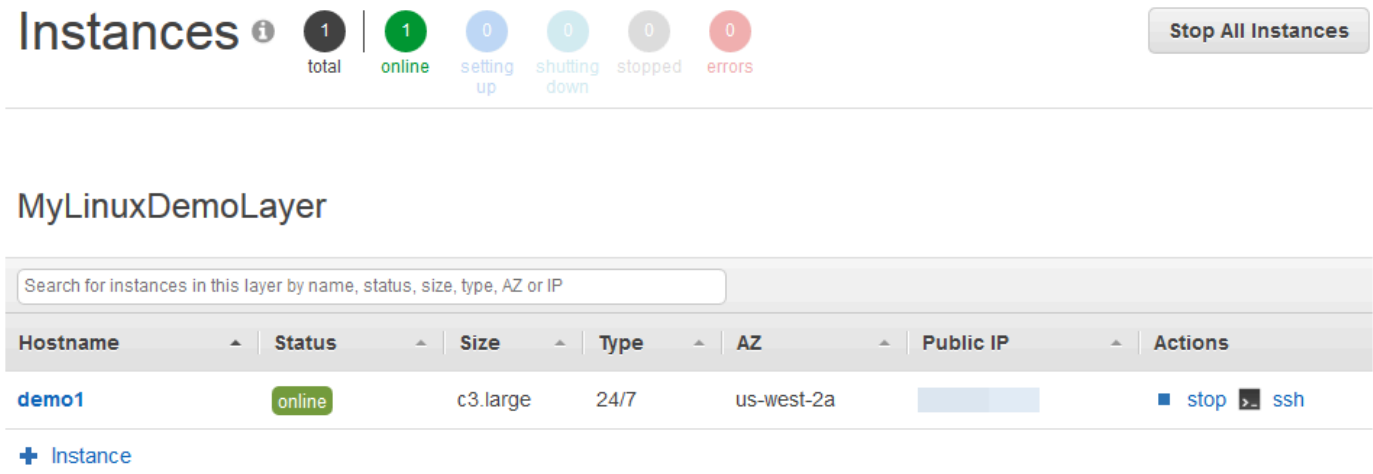
7. Pilih Tambahkan Instance. AWS OpsWorks Stacks menambahkan instance ke layer dan menampilkan halaman Instances.
8. Untuk MyLinuxDemoLayer, untuk demo1, untuk Tindakan, pilih mulai:



9. Selama beberapa menit, hal berikut terjadi:
   - Lingkaran pengaturan berubah dari 0 menjadi 1.

- Status berubah dari berhenti menjadi diminta, ke pending, ke booting, ke running_setup, dan akhirnya ke online. Perhatikan bahwa proses ini dapat memakan waktu beberapa menit.
- Setelah Status berubah menjadi online, indikator lingkaran pengaturan berubah dari 1 menjadi 0, dan lingkaran online berubah dari 0 menjadi 1 dan berubah menjadi hijau terang. Jangan lanjutkan sampai lingkaran online berubah menjadi hijau terang, dan tunjukkan 1 contoh online.

10. Hasil Anda harus sesuai dengan tangkapan layar berikut sebelum melanjutkan (jika Anda menerima pesan kegagalan, Anda mungkin ingin berkonsultasi dengan Panduan Debugging dan Pemecahan Masalah):



Anda sekarang memiliki instance yang siap untuk aplikasi yang akan digunakan untuk itu.

> **Note**
>
> Jika Anda ingin masuk ke instance untuk menjelajahinya lebih lanjut, Anda harus terlebih dahulu memberikan AWS OpsWorks Stacks informasi tentang kunci SSH publik Anda (yang dapat Anda buat dengan alat seperti ssh-keygen atau PuTTYgen), dan kemudian Anda harus mengatur izin pada `MyLinuxDemoStack` tumpukan untuk memungkinkan pengguna Anda masuk ke instance. Lihat petunjuknya di Mendaftarkan Kunci SSH Publik Pengguna dan Login dengan SSH. Jika Anda berencana menggunakan SSH untuk terhubung ke instance melalui PuTTY, lihat Menghubungkan ke Instance Linux Anda dari Windows Menggunakan PuTTY dalam dokumentasi. AWS

Pada langkah berikutnya, Anda akan menerapkan aplikasi ke instance.

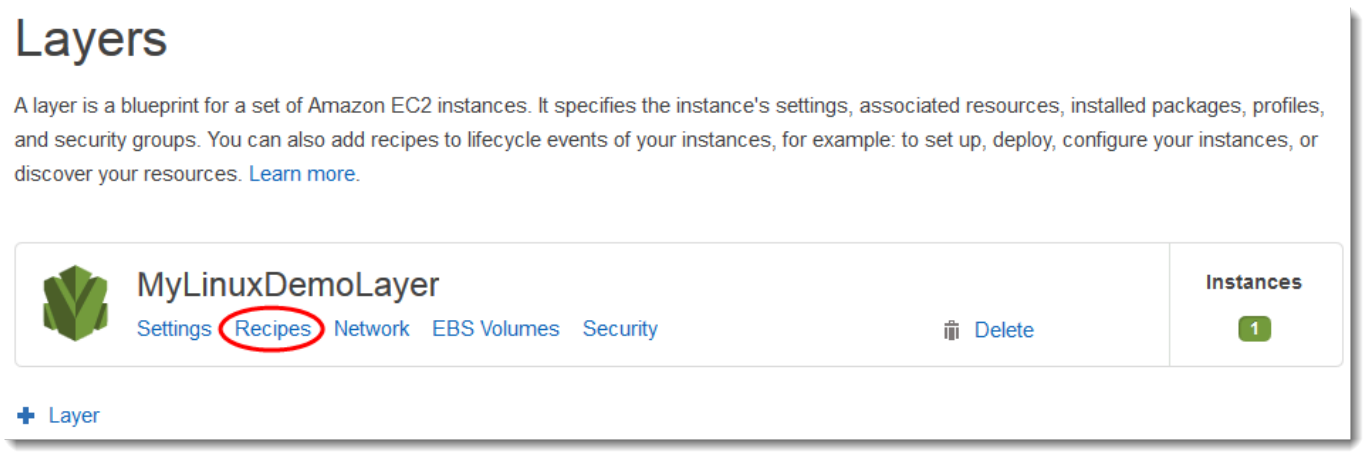
Langkah 6: Menerapkan Aplikasi ke Instans

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pada langkah ini, Anda akan menerapkan aplikasi dari GitHub ke instance yang sedang berjalan. (Untuk informasi selengkapnya, lihat Menerapkan Aplikasi.) Sebelum menerapkan aplikasi, Anda harus menentukan resep yang akan digunakan untuk mengoordinasikan penerapan. Resep adalah konsep Chef. Resep adalah instruksi, ditulis dengan sintaks bahasa Ruby, yang menentukan sumber daya yang akan digunakan dan urutan penerapan sumber daya tersebut. (Untuk informasi lebih lanjut, kunjungi Tentang Resep di situs web Learn Chef.)

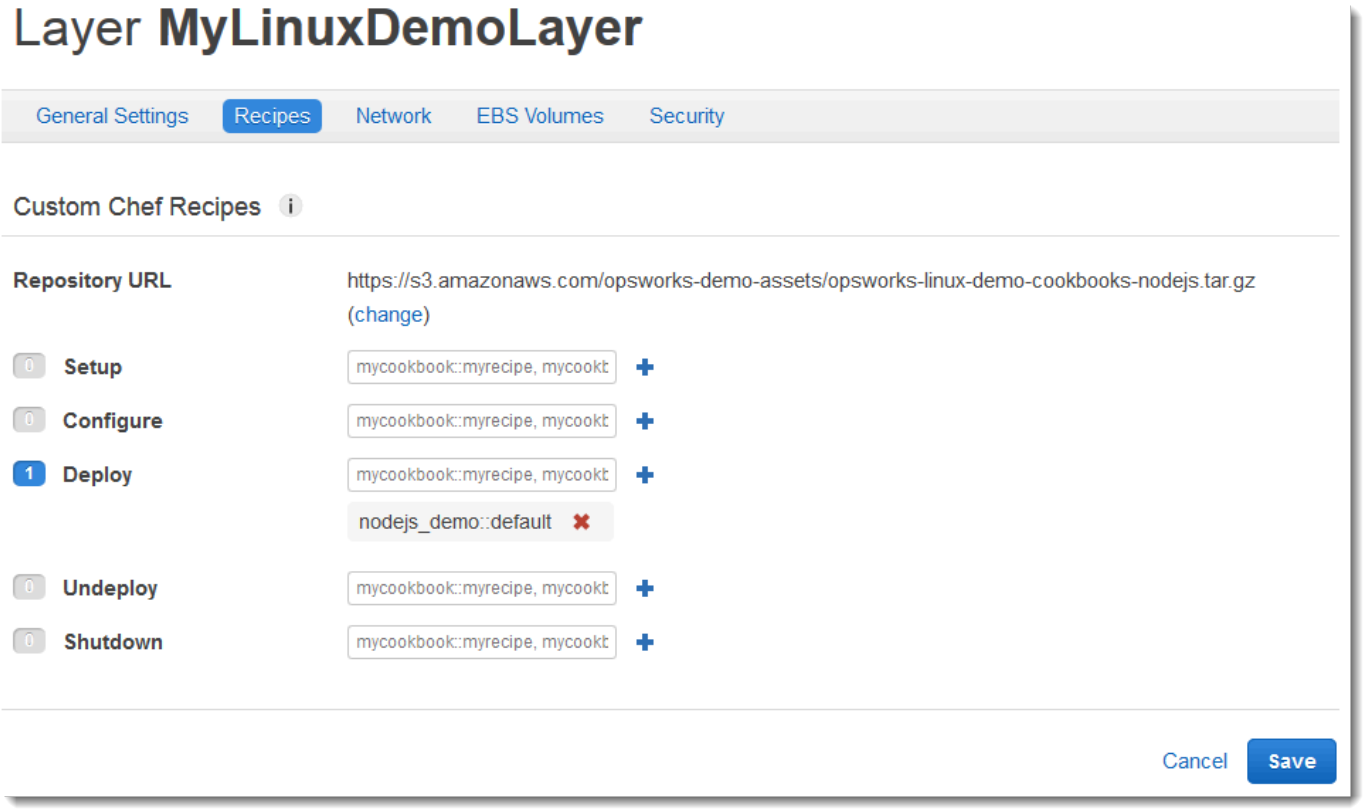
Untuk menentukan resep yang akan digunakan untuk menerapkan aplikasi ke instance

1. Di panel navigasi layanan, pilih Layers. Halaman Layers ditampilkan.
2. Untuk MyLinuxDemoLayer, pilih Resep:



   MyLinuxDemoLayerHalaman Layer ditampilkan dengan tab Recipes terbuka.
3. Untuk Resep Koki Kustom, untuk Deploy, ketik**`nodejs_demo::default`**, lalu tekan Enter. `nodejs_demo`adalah nama buku masak dan `default` merupakan nama resep target dalam

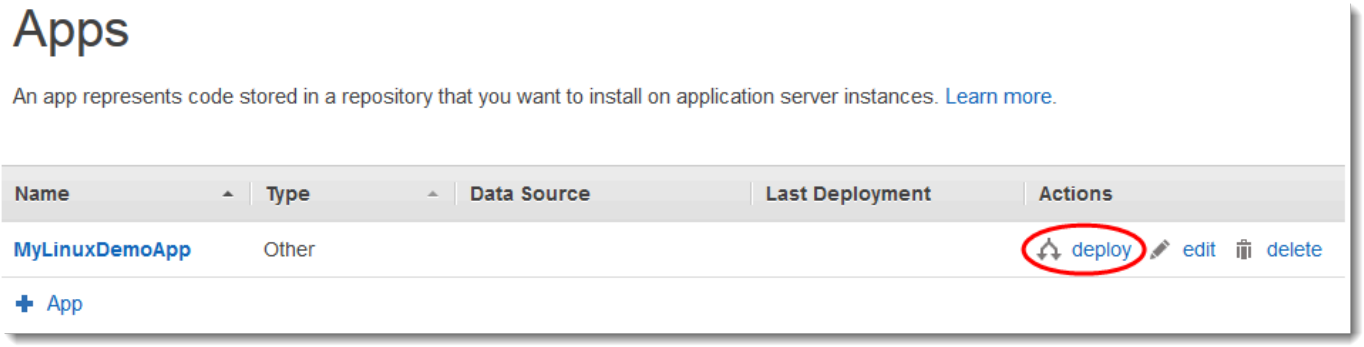
buku masak. (Untuk menjelajahi kode resep, lihatPelajari Lebih Lanjut: Jelajahi Buku Masak yang Digunakan dalam Panduan Ini.) Hasil Anda harus sesuai dengan tangkapan layar berikut:



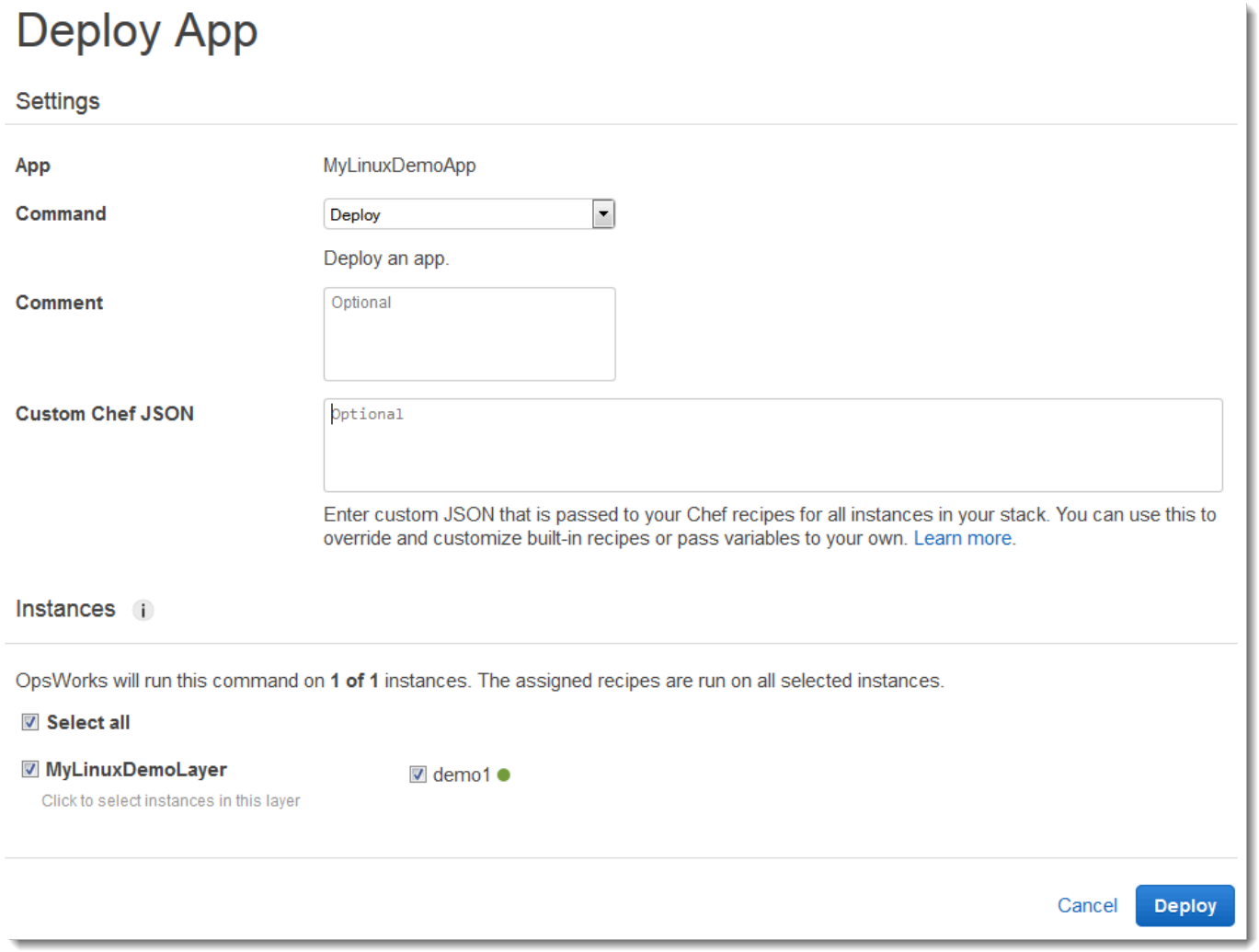
4. Pilih Simpan. AWS OpsWorks Stacks menambahkan resep ke acara siklus hidup Deploy layer.

Untuk menerapkan aplikasi ke instance

1. Di panel navigasi layanan, pilih Aplikasi. Halaman Aplikasi ditampilkan.
2. Untuk MyLinuxDemoApp, untuk Tindakan, pilih deploy, seperti yang ditampilkan dalam tangkapan layar berikut:



3. Pada halaman Deploy App, biarkan default untuk hal berikut:
   - Perintah (Menyebarkan)
   - Komentar (kosong)
   - Pengaturan, Lanjutan, Custom Chef JSON (kosong)
   - Contoh, Lanjutan (dicentang Pilih semua, diperiksa MyLinuxDemoLayer, diperiksa demo1)
4. Hasil Anda harus sesuai dengan tangkapan layar berikut:



5. Pilih Deploy. Halaman Deployment MyLinuxDemoApp — deploy ditampilkan. Status berubah dari berjalan menjadi sukses. Lingkaran berputar ditampilkan di sebelah demo1, yang kemudian berubah menjadi tanda centang hijau. Perhatikan bahwa proses ini dapat memakan waktu beberapa menit. Jangan lanjutkan sampai Anda melihat status sukses dan ikon tanda centang hijau.
6. Hasil Anda harus cocok dengan tangkapan layar berikut kecuali tentu saja untuk Dibuat pada, Selesai pada, Durasi, dan Pengguna. Jika status gagal, lalu untuk memecahkan masalah, untuk Log, pilih tampilkan untuk mendapatkan detail tentang kegagalan:



Anda sekarang telah berhasil menerapkan aplikasi ke instance.

Pada langkah berikutnya, Anda akan menguji aplikasi yang diterapkan pada instance.

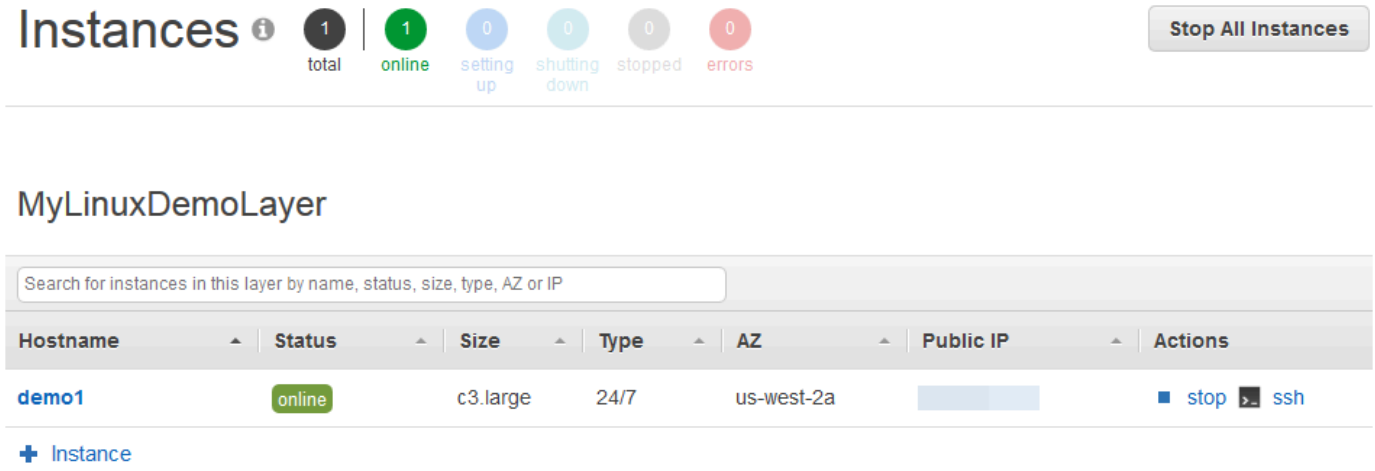
Langkah 7: Uji Aplikasi yang Diterapkan pada Instans

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang, uji penerapan aplikasi pada instance.

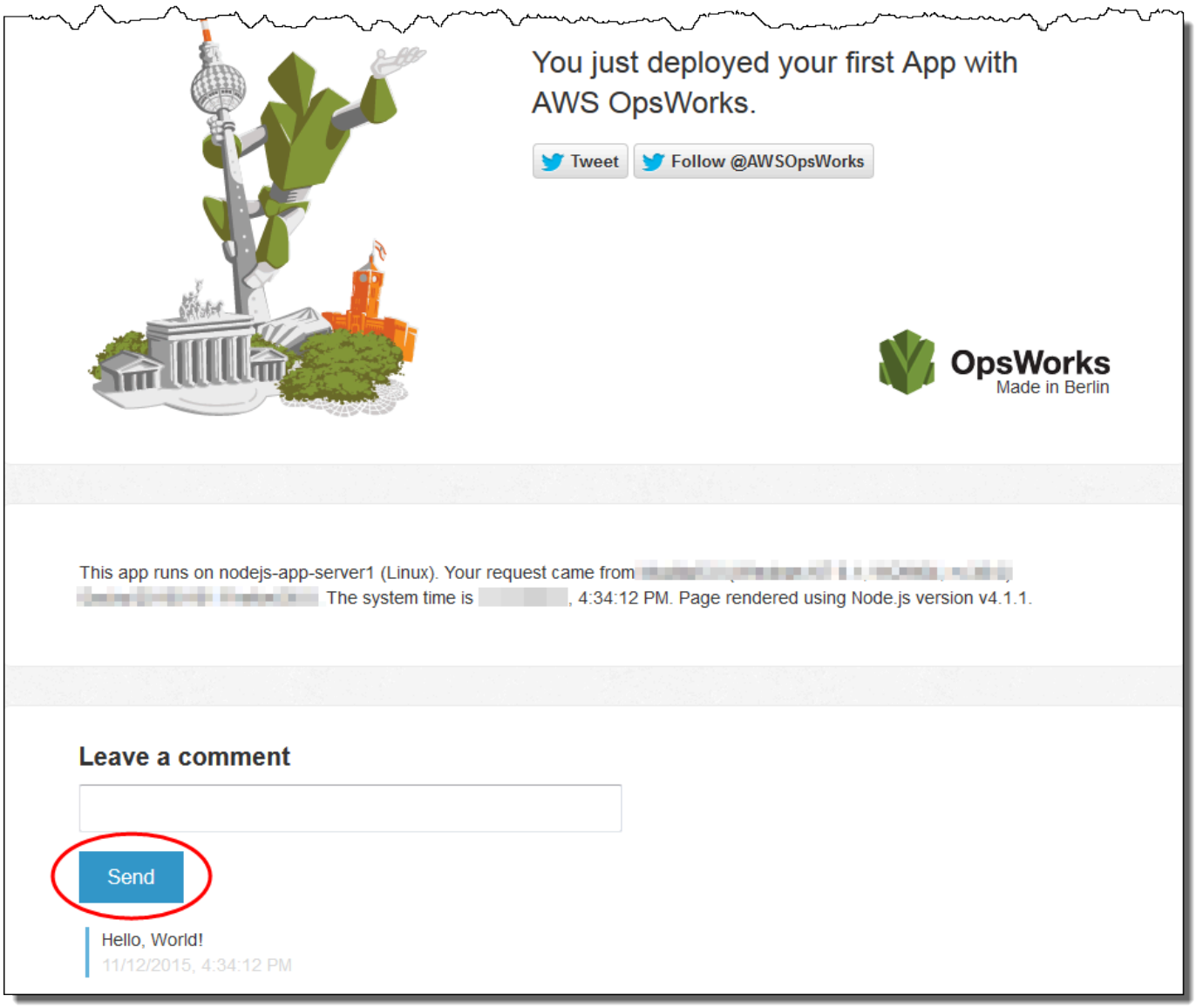
Untuk menguji penerapan pada instance

1. Di panel navigasi layanan, pilih Instans. Halaman Instans ditampilkan.
2. Untuk MyLinuxDemoLayer, untuk demo1, untuk IP Publik, pilih alamat IP:



Tab browser web baru menampilkan aplikasi.

3. Di halaman web ucapan selamat, di kotak teks Tinggalkan komentar, ketik komentar, lalu pilih Kirim untuk menguji aplikasi. Aplikasi menambahkan komentar Anda ke halaman web. Lanjutkan meninggalkan komentar dan memilih Kirim sesering yang Anda inginkan:



4. Jika Anda memiliki akun Twitter, pilih Tweet atau Ikuti @ AWSOps Works, dan ikuti petunjuk di layar untuk men-tweet tentang aplikasi atau mengikuti @ AWSOps Works.

Anda sekarang telah berhasil menguji aplikasi yang diterapkan pada instance.

Pada langkah selanjutnya, Anda dapat membersihkan AWS sumber daya yang Anda gunakan untuk panduan ini. Langkah selanjutnya adalah opsional. Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks. Namun, menyimpan AWS sumber daya ini dapat mengakibatkan beberapa tagihan berkelanjutan ke AWS akun Anda. Jika Anda ingin menyimpan AWS sumber daya ini untuk digunakan nanti, Anda sekarang telah menyelesaikan panduan ini, dan Anda dapat melompat ke depan. Langkah Berikutnya

Langkah 8 (Opsional): Bersihkan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk mencegah timbulnya biaya tambahan ke AWS akun Anda, Anda dapat menghapus AWS sumber daya yang digunakan untuk panduan ini. AWS Sumber daya ini termasuk tumpukan AWS OpsWorks Stacks dan komponen tumpukan. (Untuk informasi lebih lanjut, lihat AWS OpsWorks Harga.) Namun, Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks. Jika Anda ingin menyimpan AWS sumber daya ini tersedia, Anda sekarang telah menyelesaikan panduan ini, dan Anda dapat melompat ke. Langkah Berikutnya

Konten yang disimpan dalam sumber daya yang Anda buat untuk panduan ini dapat berisi informasi identifikasi pribadi. Jika Anda tidak lagi ingin informasi ini disimpan AWS, ikuti langkah-langkah dalam topik ini.

Untuk menghapus aplikasi dari tumpukan

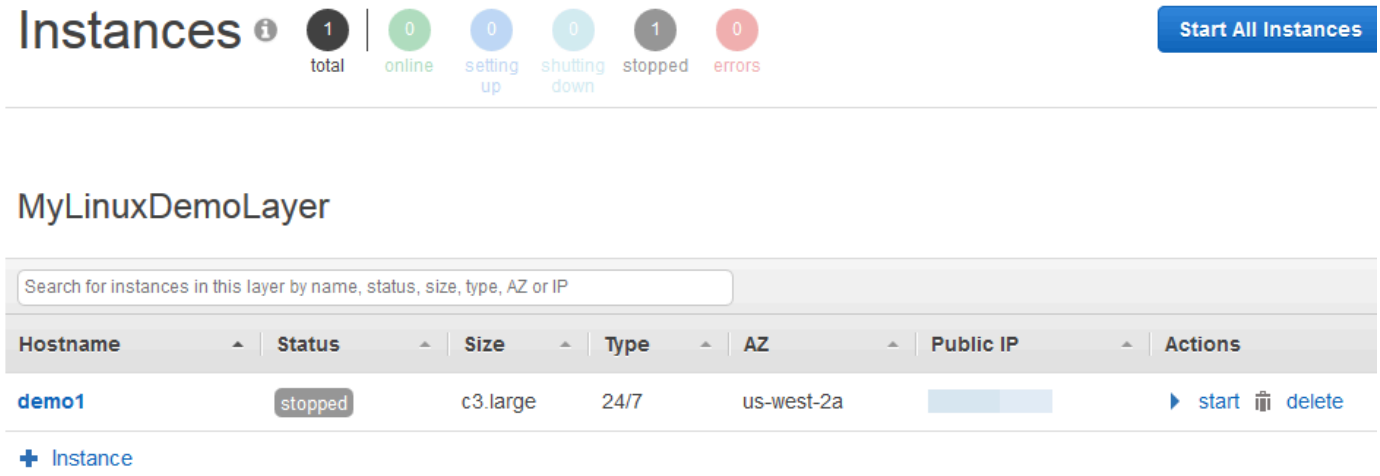
1. Di konsol AWS OpsWorks Stacks, di panel navigasi layanan, pilih Aplikasi. Halaman Aplikasi ditampilkan.
2. Untuk MyLinuxDemoApp, untuk Tindakan, pilih hapus. Saat pesan konfirmasi ditampilkan, pilih Hapus. AWS OpsWorks Stacks menghapus aplikasi.

Untuk menghapus instance untuk tumpukan

1. Di panel navigasi layanan, pilih Instans. Halaman Instans ditampilkan.
2. Untuk MyLinuxDemoLayer, untuk demo1, untuk Tindakan, pilih berhenti. Saat Anda melihat pesan konfirmasi, pilih Berhenti. Berikut ini terjadi.
   - Status berubah dari online menjadi berhenti, dan akhirnya berhenti.
   - perubahan online dari 1 menjadi 0.

- mematikan perubahan dari 0 ke 1, dan akhirnya kembali ke 0.
- berhenti akhirnya berubah dari 0 ke 1.

Proses ini dapat menghabiskan waktu beberapa menit. Ketika AWS OpsWorks Stacks selesai, hasil berikut akan ditampilkan.



3. Untuk Tindakan, pilih hapus. Saat Anda melihat pesan konfirmasi, pilih Hapus. AWS OpsWorks Stacks menghapus instance dan menampilkan pesan No instance.

Untuk menghapus tumpukan

1. Di panel navigasi layanan, pilih Stack. Halaman MyLinuxDemoStack ditampilkan.
2. Pilih Hapus Tumpukan. Saat Anda melihat pesan konfirmasi, pilih Hapus. AWS OpsWorks Stacks menghapus tumpukan dan menampilkan halaman OpsWorks Dashboard.

Secara opsional, Anda dapat menghapus user dan Amazon EC2 key pair yang Anda gunakan untuk panduan ini, jika Anda tidak ingin menggunakannya kembali untuk akses ke layanan dan instance lain AWS . EC2 Untuk petunjuk, lihat Menghapus pengguna IAM dan pasangan EC2 kunci Amazon dan instance Linux.

Anda sekarang telah menyelesaikan panduan ini. Untuk informasi selengkapnya, lihat Langkah Berikutnya.

Langkah Berikutnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang setelah Anda menyelesaikan panduan ini, Anda dapat mempelajari lebih lanjut tentang menggunakan AWS OpsWorks Stacks:

- Jelajahi buku masak dan aplikasi yang Anda gunakan untuk panduan ini. Lihat Pelajari Lebih Lanjut: Jelajahi Buku Masak yang Digunakan dalam Panduan Ini dan Pelajari Lebih Lanjut: Jelajahi Aplikasi yang Digunakan dalam Panduan Ini.
- Berlatih menggunakan AWS OpsWorks Stacks dengan instance Windows. Lihat Memulai: Windows.
- Pelajari lebih lanjut tentang tumpukan dengan mempelajari caranyaMembuat Tumpukan Baru.
- Pelajari lebih lanjut tentang layer byMengedit Konfigurasi OpsWorks Layer.
- Pelajari lebih lanjut tentang instans olehMenambahkan Instance ke Layer.
- Pelajari lebih lanjut tentang aplikasi olehMenerapkan Aplikasi.
- Pelajari lebih lanjut tentang Buku Masak dan Resep.
- Buat buku masak Anda sendiri. Lihat Memulai: Buku Masak.
- Pelajari tentang mengontrol akses ke tumpukan dengan Keamanan dan Izin.

Pelajari Lebih Lanjut: Jelajahi Buku Masak yang Digunakan dalam Panduan Ini

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

> Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini menjelaskan buku masak yang digunakan AWS OpsWorks Stacks untuk penelusuran.

Buku masak adalah konsep Chef. Buku masak adalah file arsip yang berisi informasi konfigurasi, seperti resep, nilai atribut, file, templat, pustaka, definisi, dan sumber daya khusus. Resep juga merupakan konsep Chef. Resep adalah instruksi, ditulis dengan sintaks bahasa Ruby, yang menentukan sumber daya yang akan digunakan dan urutan penerapan sumber daya tersebut. Untuk informasi lebih lanjut, kunjungi Tentang Buku Masak dan Tentang Resep di situs web Learn Chef.

Untuk melihat isi buku masak yang digunakan dalam panduan ini, ekstrak isi file opsworks-linux-demo-cookbooks-nodejs.tar.gz ke direktori kosong di workstation lokal Anda. (Anda juga dapat masuk ke instance tempat Anda menggunakan buku masak dan menjelajahi konten `/var/chef/cookbooks` direktori.)

`default.rb`File dalam `cookbooks/nodejs_demo/recipes` direktori adalah tempat buku masak menjalankan kodenya:

```
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
```

Inilah yang dilakukan file tersebut:

- `search(:aws_opsworks_app).first`menggunakan pencarian Chef untuk mencari informasi tentang aplikasi yang pada akhirnya akan digunakan ke instance. Informasi ini mencakup pengaturan seperti nama pendek aplikasi dan detail repositori sumbernya. Karena hanya satu aplikasi yang diterapkan dalam panduan ini, pencarian Chef mendapatkan pengaturan ini dari item informasi pertama dalam indeks `aws_opsworks_app` pencarian pada instance. Setiap kali sebuah instance diluncurkan, AWS OpsWorks Stacks menyimpan ini dan informasi terkait lainnya sebagai sekumpulan kantong data pada instance itu sendiri, dan Anda mendapatkan konten kantong data melalui pencarian Chef. Meskipun Anda dapat membuat kode keras pengaturan ini ke dalam resep ini, menggunakan tas data dan pencarian Chef adalah pendekatan yang lebih kuat. Untuk informasi selengkapnya tentang tas data, lihatAWS OpsWorks Referensi Tas Data Stacks. Lihat juga Tentang Tas Data di situs web Learn Chef. Untuk informasi lebih lanjut tentang pencarian Chef, buka Tentang Pencarian di situs web Learn Chef.
- Sumber `package` daya menginstal Git pada instance.
- Sumber `application` daya menjelaskan dan menyebarkan aplikasi web:
  - `javascript`adalah versi JavaScript runtime yang akan diinstal.
  - `environment`menetapkan variabel lingkungan.
  - `git`mendapatkan kode sumber dari repositori dan cabang yang ditentukan.
  - `app_path`adalah jalur untuk mengkloning repositori ke. Jika jalur tidak ada pada instance, AWS OpsWorks Stacks membuatnya.
  - `link`membuat tautan simbolis.
  - `npm_install`menginstal Node Package Manager, manajer paket default untuk Node.js.
  - `npm_start`menjalankan Node.js.

Meskipun AWS OpsWorks Stacks membuat buku masak yang digunakan untuk panduan ini, Anda dapat membuat buku masak sendiri. Untuk mempelajari caranya, lihat Memulai: Buku Masak. Juga, buka Tentang Buku Masak, Tentang Resep, dan Pelajari Dasar-Dasar Koki di Ubuntu di situs web Learn Chef, dan bagian “Buku masak Chef pertama kami” di Langkah pertama dengan Chef di situs web Memulai dengan Chef.

Pelajari Lebih Lanjut: Jelajahi Aplikasi yang Digunakan dalam Panduan Ini

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini menjelaskan aplikasi yang diterapkan AWS OpsWorks Stacks ke instance untuk panduan ini.

Untuk melihat kode sumber aplikasi, ekstrak konten opsworks-windows-demo-nodejs GitHub repositori ke direktori kosong di workstation lokal Anda. Anda juga dapat masuk ke instance tempat Anda menggunakan buku masak dan menjelajahi isi direktori. `/srv/mylinuxdemoapp`

`index.js`File berisi kode paling signifikan untuk aplikasi:

```
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);
});
```

Inilah yang dilakukan file tersebut:

- `require`memuat modul yang berisi beberapa kode dependen yang perlu dijalankan aplikasi web ini seperti yang diharapkan.
- `get_comments`Fungsi `add_comment` dan menulis informasi ke, dan membaca informasi dari, `comments.json` file.

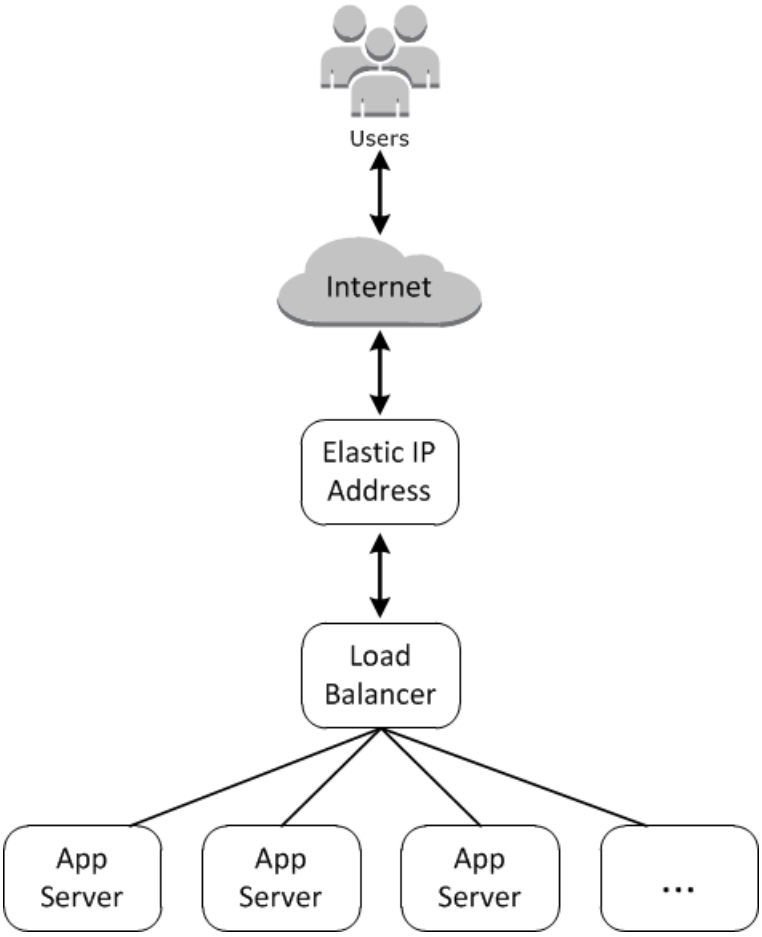
- Untuk informasi tentang`app.get`,,`app.listen`,`app.post`,`app.set`, dan`app.use`, lihat Referensi API Ekspres.

Untuk mempelajari cara membuat dan mengemas aplikasi Anda untuk penerapan, lihatSumber Aplikasi.

Memulai dengan Windows Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Aplikasi berbasis cloud biasanya memerlukan sekelompok sumber daya terkait — server aplikasi, server basis data, dan sebagainya — yang harus dibuat dan dikelola secara kolektif. Kumpulan instance ini disebut tumpukan. Tumpukan aplikasi sederhana mungkin terlihat seperti berikut ini.



Arsitektur dasar terdiri dari yang berikut:

- Alamat IP Elastis untuk menerima permintaan pengguna.
- Load balancer untuk mendistribusikan permintaan masuk secara merata di seluruh server aplikasi.
- Satu set instance server aplikasi, sebanyak yang diperlukan untuk menangani lalu lintas.

Selain itu, Anda biasanya memerlukan cara untuk mendistribusikan aplikasi ke server aplikasi, mengelola izin pengguna, dan sebagainya.

AWS OpsWorks Stacks menyediakan cara sederhana dan mudah untuk membuat dan mengelola tumpukan dan aplikasi serta sumber daya terkait. Bab ini memperkenalkan dasar-dasar AWS OpsWorks tumpukan — bersama dengan beberapa fitur yang lebih canggih — dengan memandu Anda melalui proses pembuatan tumpukan server aplikasi dalam diagram. Ini menggunakan model pengembangan inkremental yang memudahkan AWS OpsWorks Stacks untuk diikuti: Siapkan tumpukan dasar dan, setelah berfungsi dengan benar, tambahkan komponen hingga Anda tiba pada implementasi berfitur lengkap.

- Langkah 1: Lengkapi Prasyaratmenunjukkan cara mengatur untuk memulai penelusuran.
- Langkah 2: Buat Stack Server Aplikasi Dasarmenunjukkan cara membuat tumpukan dasar untuk mendukung Internet Information Services (IIS) dan menyebarkan aplikasi ke server.
- Langkah 3: Skala Keluar IISExamplemenunjukkan cara menskalakan tumpukan untuk menangani peningkatan beban dengan menambahkan lebih banyak server aplikasi, penyeimbang beban untuk mendistribusikan lalu lintas masuk, dan alamat IP Elastis untuk menerima permintaan yang masuk.

Topik

- Langkah 1: Lengkapi Prasyarat
- Langkah 2: Buat Stack Server Aplikasi Dasar
- Langkah 3: Skala Keluar IISExample
- Langkah Berikutnya

Langkah 1: Lengkapi Prasyarat

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Selesaikan langkah-langkah pengaturan berikut sebelum Anda dapat memulai penelusuran. Langkah-langkah penyiapan ini termasuk mendaftar untuk AWS akun, membuat pengguna administratif, dan menetapkan izin akses ke Stacks. AWS OpsWorks

Jika Anda telah menyelesaikan Memulai: Contoh atau Memulai: Linux penelusuran, maka Anda telah memenuhi prasyarat untuk panduan ini, dan Anda dapat melompat ke depan. Langkah 2: Buat Stack Server Aplikasi Dasar

Topik

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif

- Tetapkan Izin Akses Layanan
- Pastikan Pengguna AWS OpsWorks Stacks Ditambahkan ke Domain Anda

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Tetapkan Izin Akses Layanan

Aktifkan akses ke layanan AWS OpsWorks Stacks (dan layanan terkait yang diandalkan oleh AWS OpsWorks Stacks) dengan menambahkan `AWSOpsWorks_FullAccess` dan `AmazonS3FullAccess` izin ke peran atau pengguna Anda.

Untuk informasi selengkapnya tentang menambahkan izin, lihat Menambahkan izin identitas IAM (konsol).

Pastikan Pengguna AWS OpsWorks Stacks Ditambahkan ke Domain Anda

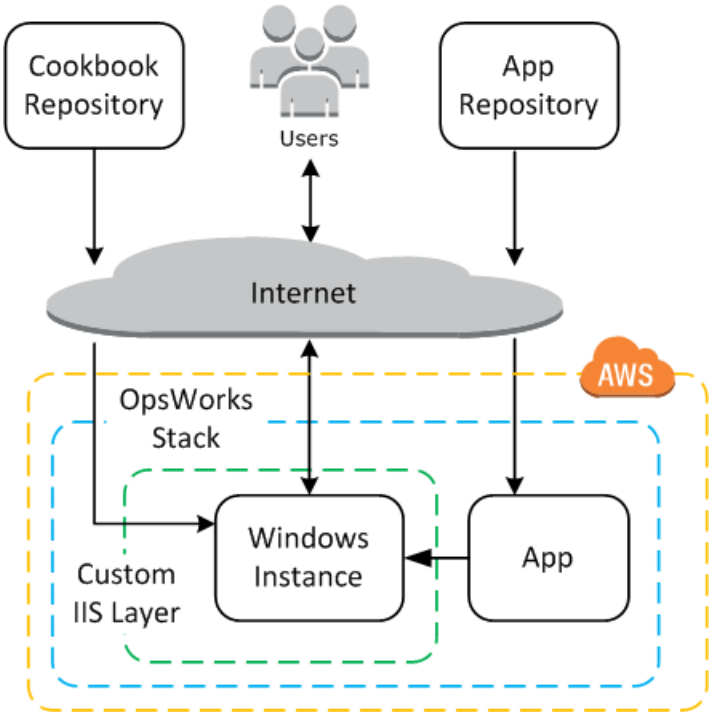
Dalam tumpukan Chef 12.2, `aws_opsworks_users` buku masak yang disertakan membuat pengguna yang memiliki akses SSH dan Remote Desktop Protocol (RDP) ke instance berbasis Windows. Saat Anda menggabungkan instance Windows di tumpukan Anda ke domain Active Directory, buku masak ini dapat gagal jika pengguna AWS OpsWorks Stacks tidak ada di Active Directory. Jika pengguna tidak dikenali di Active Directory, instance dapat memasukkan `setup failed` status saat Anda memulai ulang setelah bergabung dengan mereka ke domain. Untuk instance Windows yang bergabung dengan domain, tidak cukup untuk memberikan SSH/RDP akses kepada pengguna AWS OpsWorks Stacks di halaman izin pengguna.

Sebelum Anda menggabungkan instance Windows di tumpukan Chef 12.2 ke domain Active Directory, pastikan semua pengguna AWS OpsWorks Stacks dari tumpukan berbasis Windows adalah anggota domain. Cara terbaik untuk melakukannya adalah dengan mengonfigurasi identitas federasi dengan IAM sebelum membuat tumpukan berbasis Windows Anda, dan kemudian mengimpor pengguna federasi ke AWS OpsWorks Stacks sebelum menggabungkan instance di tumpukan Anda ke domain. Untuk informasi selengkapnya tentang cara melakukannya, lihat Mengaktifkan Federasi ke AWS Menggunakan Windows Active Directory, ADFS, dan SAMP 2.0 di AWS Security Blog, dan Menggabungkan Pengguna yang Ada di Panduan Pengguna IAM.

Langkah 2: Buat Stack Server Aplikasi Dasar

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan server aplikasi dasar terdiri dari satu instance server aplikasi dengan alamat IP publik untuk menerima permintaan pengguna. Kode aplikasi dan file terkait disimpan dalam repositori terpisah dan digunakan dari sana ke server. Diagram berikut menggambarkan tumpukan seperti itu.



Tumpukan memiliki komponen berikut:

- Lapisan, yang mewakili sekelompok instance dan menentukan bagaimana mereka akan dikonfigurasi.

  Lapisan dalam contoh ini mewakili sekelompok contoh IIS.
- Sebuah instance, yang mewakili EC2 instance Amazon.

  Dalam hal ini, lapisan mengkonfigurasi satu instance untuk menjalankan IIS, tetapi lapisan dapat memiliki sejumlah instance.
- Aplikasi, yang berisi informasi yang diperlukan untuk menginstal aplikasi pada instance.
- Buku masak, yang berisi resep Chef khusus yang mendukung lapisan IIS khusus. Buku masak dan kode aplikasi disimpan di repositori jarak jauh, seperti file arsip di bucket Amazon S3 atau repositori Git.

Bagian berikut menjelaskan cara menggunakan konsol AWS OpsWorks Stacks untuk membuat tumpukan dan menyebarkan aplikasi.

Topik

- Langkah 2.1: Buat Stack
- Langkah 2.2: Otorisasi Akses RDP
- Langkah 2.3: Menerapkan Buku Masak Kustom
- Langkah 2.4: Tambahkan Layer IIS
- Langkah 2.5: Menerapkan Aplikasi

Langkah 2.1: Buat Stack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda memulai proyek AWS OpsWorks Stacks dengan membuat tumpukan, yang bertindak sebagai wadah untuk instance Anda dan sumber daya lainnya. Konfigurasi tumpukan menentukan beberapa pengaturan dasar, seperti wilayah AWS dan sistem operasi default, yang dibagikan oleh semua instance tumpukan.

Untuk membuat tumpukan baru

1. Tambahkan Stack

   Jika Anda belum melakukannya, masuk ke konsol AWS OpsWorks Stacks.

   - Jika akun tidak memiliki tumpukan yang ada, Anda akan melihat OpsWorks halaman Selamat Datang di AWS; pilih Tambahkan tumpukan pertama Anda.
   - Jika tidak, Anda akan melihat dasbor AWS OpsWorks Stacks, yang mencantumkan tumpukan akun Anda; pilih Tambahkan Tumpukan.

2. Konfigurasikan Stack

   Pada halaman Add Stack, pilih tumpukan Chef 12 dan tentukan pengaturan berikut:

Nama tumpukan

Masukkan nama untuk tumpukan Anda, yang dapat berisi karakter alfanumerik (a—z, A—Z, dan 0-9), dan tanda hubung (-). Contoh tumpukan untuk panduan ini diberi nama. **`IISWalkthrough`**

Wilayah

Pilih US West (Oregon) sebagai wilayah tumpukan.

Anda dapat membuat tumpukan di wilayah mana pun, tetapi kami merekomendasikan US West (Oregon) untuk tutorial.

Sistem operasi default

Pilih Windows, lalu tentukan Microsoft Windows Server 2022 Base, yang merupakan pengaturan default.

Gunakan buku masak Chef khusus

Untuk keperluan panduan ini, tentukan Tidak untuk opsi ini.

3. Pilih Advanced untuk mengonfirmasi bahwa Anda memiliki peran IAM dan profil instans IAM default yang dipilih.

   Peran IAM

   Tentukan peran IAM (AWS Identity and Access Management) stack. AWS OpsWorks Stacks perlu mengakses layanan AWS lainnya untuk melakukan tugas-tugas seperti membuat dan mengelola EC2 instans Amazon. Peran IAM menentukan peran yang diasumsikan AWS OpsWorks Stacks untuk bekerja dengan layanan AWS lain atas nama Anda. Untuk informasi selengkapnya, lihat Mengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda.

   - Jika akun Anda memiliki peran IAM AWS OpsWorks Stacks yang ada, Anda dapat memilihnya dari daftar.

     Jika peran itu dibuat oleh AWS OpsWorks Stacks, itu akan diberi nama`aws-opsworks-service-role`.

   - Jika tidak, pilih Peran IAM Baru untuk mengarahkan AWS OpsWorks Tumpukan untuk membuat peran baru bagi Anda dengan izin yang benar.

Catatan: Jika Anda memiliki izin Akses Penuh AWS OpsWorks Tumpukan, membuat peran baru memerlukan beberapa izin IAM tambahan. Untuk informasi selengkapnya, lihat Contoh kebijakan.

4. Terima nilai default untuk pengaturan lain dan pilih Tambah Tumpukan. Untuk informasi lebih lanjut tentang berbagai pengaturan tumpukan, lihatMembuat Tumpukan Baru.

Langkah 2.2: Otorisasi Akses RDP

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang Anda telah membuat tumpukan, Anda akan membuat layer dan menambahkan instance Windows ke layer. Namun, sebelum Anda dapat melakukannya, Anda harus mengkonfigurasi tumpukan untuk memungkinkan Anda menggunakan RDP untuk terhubung ke instance lapisan kustom. Untuk melakukannya, Anda harus melakukan hal berikut:

- Tambahkan aturan masuk ke grup keamanan yang mengontrol akses RDP.
- Tetapkan izin AWS OpsWorks Stacks Anda untuk tumpukan ini untuk memungkinkan akses RDP.

Saat Anda membuat tumpukan pertama di suatu wilayah, AWS OpsWorks Stacks membuat satu set grup keamanan. Mereka termasuk satu bernama sesuatu seperti`AWS-OpsWorks-RDP-Server`, yang AWS OpsWorks Stacks melampirkan ke semua instance Windows untuk memungkinkan akses RDP. Namun, secara default, grup keamanan ini tidak memiliki aturan apa pun, jadi Anda harus menambahkan aturan masuk untuk mengizinkan akses RDP ke instans Anda.

Untuk memungkinkan akses RDP

1. Buka EC2 konsol Amazon, atur ke wilayah tumpukan, dan pilih Grup Keamanan dari panel navigasi.
2. Pilih AWS- OpsWorks -RDP-Server, pilih tab Inbound, dan pilih Edit.

3. Pilih Tambahkan Aturan dan tentukan pengaturan berikut:

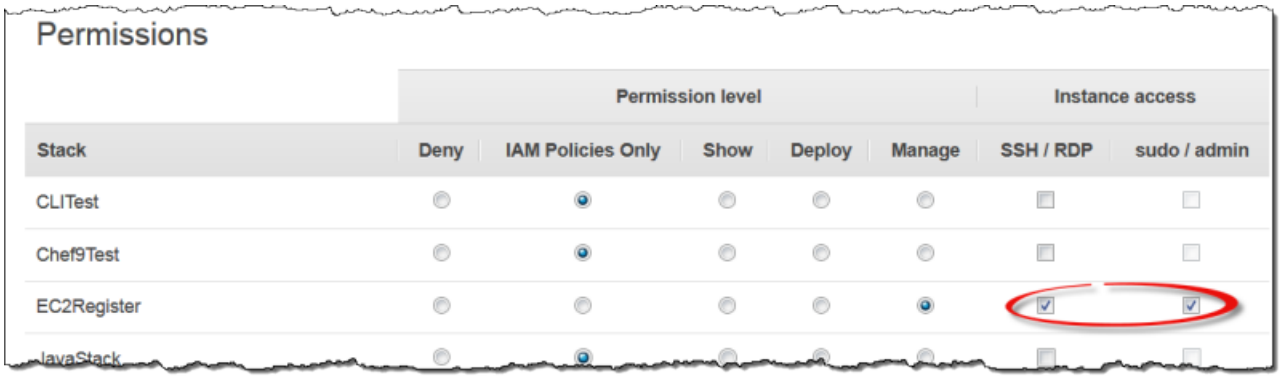
   - Jenis - RDP.
   - Sumber — Alamat IP sumber yang diizinkan.

     Anda biasanya mengizinkan permintaan RDP masuk dari alamat IP Anda atau rentang alamat IP tertentu (biasanya rentang alamat IP perusahaan Anda). Untuk tujuan pembelajaran, seringkali cukup untuk menentukan 0.0.0.0/0, yang memungkinkan akses RDP dari alamat IP apa pun.

Grup keamanan memungkinkan instance untuk menerima permintaan koneksi RDP, tetapi itu hanya setengah dari cerita. Pengguna biasa akan masuk ke instance menggunakan kata sandi yang disediakan oleh AWS OpsWorks Stacks. Agar AWS OpsWorks Stacks menghasilkan kata sandi itu, Anda harus secara eksplisit mengotorisasi akses RDP untuk pengguna.

Untuk mengotorisasi RDP untuk pengguna

1. Di dasbor AWS OpsWorks Stacks, pilih IISWalkthroughtumpukan.
2. Di panel navigasi untuk tumpukan, pilih Izin.
3. Pada halaman Izin, pilih Edit.
4. Dalam daftar pengguna, pilih kotak centang untuk SSH/RDP untuk pengguna yang ingin Anda berikan izin yang diperlukan. Jika Anda ingin pengguna juga memiliki izin administrator, pilih sudo/admin juga.



5. Pilih Simpan.

Pengguna kemudian bisa mendapatkan kata sandi dan menggunakannya untuk masuk ke instance, seperti yang dijelaskan nanti.

> **Note**
>
> Anda juga dapat masuk sebagai Administrator. Untuk informasi selengkapnya, lihat Masuk Sebagai Administrator.

Langkah 2.3: Menerapkan Buku Masak Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Meskipun tumpukan pada dasarnya adalah wadah untuk instance, Anda tidak menambahkan instance langsung ke tumpukan. Anda menambahkan satu atau lebih lapisan, yang masing-masing mewakili sekelompok instance terkait, dan kemudian menambahkan instance ke lapisan.

Lapisan pada dasarnya adalah cetak biru yang digunakan AWS OpsWorks Stacks untuk membuat satu set EC2 instance Amazon dengan konfigurasi yang sama. Sebuah instance dimulai dengan versi dasar sistem operasi, dan lapisan instance melakukan berbagai tugas pada instance untuk mengimplementasikan cetak biru itu, yang dapat mencakup:

- Membuat direktori dan file
- Mengelola pengguna
- Menginstal dan mengkonfigurasi perangkat lunak
- Memulai atau menghentikan server
- Menyebarkan kode aplikasi dan file terkait.

Lapisan melakukan tugas pada instance dengan menjalankan resep Chef —resep singkatnya. Resep adalah aplikasi Ruby yang menggunakan bahasa khusus domain Chef (DSL) untuk menggambarkan keadaan akhir dari instance. Dengan AWS OpsWorks Stacks, setiap resep biasanya ditetapkan ke salah satu peristiwa siklus hidup layer: Setup, Configuration, Deploy, Undeploy, dan Shutdown. Ketika peristiwa siklus hidup terjadi pada sebuah instance, AWS OpsWorks Stacks menjalankan

resep acara untuk melakukan tugas yang sesuai. Misalnya, peristiwa Setup terjadi setelah instance selesai booting. AWS OpsWorks Stacks kemudian menjalankan resep Setup, yang biasanya melakukan tugas-tugas seperti menginstal dan mengkonfigurasi perangkat lunak server dan memulai layanan terkait.

AWS OpsWorks Stacks menyediakan setiap lapisan dengan satu set resep bawaan yang melakukan tugas standar. Anda dapat memperluas fungsionalitas lapisan dengan menerapkan resep khusus untuk melakukan tugas tambahan dan menetapkannya ke peristiwa siklus hidup lapisan. Tumpukan Windows mendukung lapisan khusus, yang memiliki seperangkat resep minimal yang hanya melakukan beberapa tugas dasar. Untuk menambahkan fungsionalitas ke instance Windows Anda, Anda harus menerapkan resep khusus untuk menginstal perangkat lunak, menyebarkan aplikasi, dan sebagainya. Topik ini menjelaskan cara membuat lapisan kustom sederhana untuk mendukung instance IIS.

Topik

- Pengantar Cepat untuk Buku Masak dan Resep
- Menerapkan Resep untuk Menginstal dan Memulai IIS
- Aktifkan Buku Masak Kustom

Pengantar Cepat untuk Buku Masak dan Resep

Sebuah resep mendefinisikan satu atau lebih aspek dari status yang diharapkan suatu instans: direktori apa yang harus dimiliki, paket perangkat lunak apa yang harus diinstal, aplikasi apa yang harus digunakan, dan sebagainya. Resep dikemas dalam buku masak, yang biasanya berisi satu atau lebih resep terkait, ditambah file terkait seperti template untuk membuat file konfigurasi.

Topik ini adalah pengantar resep yang sangat mendasar, cukup untuk menunjukkan kepada Anda bagaimana menerapkan buku masak untuk mendukung lapisan IIS kustom sederhana. Untuk pengenalan buku masak yang lebih umum, lihatBuku Masak dan Resep. Untuk pengantar tutorial terperinci untuk mengimplementasikan buku masak, termasuk beberapa topik khusus Windows, lihat. Buku masak 101

Resep koki secara teknis adalah aplikasi Ruby, tetapi sebagian besar, jika tidak semua, kodenya ada di Chef DSL. DSL sebagian besar terdiri dari sekumpulan sumber daya, yang dapat Anda gunakan untuk secara deklaratif menentukan aspek status instance. Misalnya, `directory`sumber daya mendefinisikan direktori yang akan ditambahkan ke sistem. Contoh berikut mendefinisikan `C:\data` direktori dengan hak kontrol penuh milik pengguna tertentu dan tidak mewarisi hak dari direktori induk.

```
directory 'C:\data' do
  rights :full_control, 'WORKGROUP\username'
  inherits false
  action :create
end
```

Ketika Chef menjalankan resep, ia mengeksekusi setiap sumber daya dengan meneruskan data ke penyedia terkait, objek Ruby yang menangani detail memodifikasi status instance. Dalam hal ini, penyedia membuat direktori baru dengan konfigurasi yang ditentukan.

Buku masak khusus untuk lapisan IIS kustom harus melakukan tugas-tugas berikut:

- Instal fitur IIS dan mulai layanan.

  Anda biasanya melakukan tugas ini selama penyiapan, tepat setelah instance selesai booting.
- Menyebarkan aplikasi ke instance, halaman HTML sederhana untuk contoh ini.

  Anda biasanya melakukan tugas ini selama pengaturan. Namun, aplikasi biasanya perlu diperbarui secara berkala, jadi Anda juga perlu menerapkan pembaruan saat instans sedang online.

Anda bisa memiliki satu resep untuk melakukan semua tugas ini. Namun, pendekatan yang lebih disukai adalah memiliki resep terpisah untuk tugas penyiapan dan penerapan. Dengan begitu, Anda dapat menerapkan pembaruan aplikasi kapan saja tanpa juga menjalankan kode penyiapan. Berikut ini menjelaskan cara menyiapkan buku masak untuk mendukung lapisan IIS kustom. Topik selanjutnya akan menunjukkan bagaimana menerapkan resep.

Untuk memulai

1. Buat direktori bernama `iis-cookbook` di lokasi yang nyaman di workstation Anda.
2. Tambahkan `metadata.rb` file dengan konten berikut ke`iis-cookbook`.

   ```
   name "iis-cookbook"
   version "0.1.0"
   ```

   Contoh ini menggunakan minimal`metadata.rb`. Untuk informasi selengkapnya tentang bagaimana Anda dapat menggunakan file ini, lihat metadata.rb.
3. Tambahkan `recipes` direktori ke`iis-cookbook`.

   Direktori ini, yang harus diberi nama`recipes`, berisi resep buku masak.

Secara umum, buku masak dapat berisi berbagai direktori lainnya. Misalnya, jika resep menggunakan templat untuk membuat file konfigurasi, templat biasanya masuk ke `templates\default` direktori. Buku masak untuk contoh ini seluruhnya terdiri dari resep, sehingga tidak memerlukan direktori lain. Juga, contoh ini menggunakan buku masak tunggal, tetapi Anda dapat menggunakan sebanyak yang Anda butuhkan; beberapa buku masak sering lebih disukai untuk proyek yang kompleks. Misalnya, Anda dapat memiliki buku masak terpisah untuk tugas penyiapan dan penerapan. Untuk contoh buku masak lainnya, lihatBuku Masak dan Resep.

Menerapkan Resep untuk Menginstal dan Memulai IIS

IIS adalah fitur Windows, salah satu dari satu set komponen sistem opsional yang dapat Anda instal di Windows Server. Anda dapat memiliki resep menginstal IIS dengan salah satu cara berikut:

- Dengan menggunakan sumber `powershell_script`daya untuk menjalankan `Install-WindowsFeature`cmdlet.
- Dengan menggunakan `windows_feature` sumber daya buku masak Chef windows.

  `windows`Buku masak berisi sekumpulan sumber daya yang penyedianya menggunakan Deployment Image Servicing and Management (DISM) untuk melakukan berbagai tugas pada instance Windows, termasuk instalasi fitur.

> **Note**
>
> `powershell_script`adalah salah satu sumber daya yang paling berguna untuk resep Windows. Anda dapat menggunakannya untuk melakukan berbagai tugas pada sebuah instance dengan menjalankan PowerShell skrip atau cmdlet. Ini sangat berguna untuk tugas-tugas yang tidak didukung oleh sumber daya Chef.

Contoh ini menjalankan PowerShell skrip untuk menginstal dan memulai Web Server (IIS). `windows`Buku masak dijelaskan nanti. Untuk contoh cara menggunakan `windows_feature` untuk menginstal IIS, lihatMenginstal Fitur Windows: IIS.

Tambahkan resep bernama `install.rb` dengan konten berikut ke `recipes` direktori buku masak.

```
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
```

Resepnya berisi dua sumber daya.

powershell_script

`powershell_script`menjalankan PowerShell skrip atau cmdlet yang ditentukan. Contoh ini memiliki pengaturan atribut berikut:

- `code`— PowerShell Cmdlet untuk dijalankan.

  Contoh ini menjalankan `Install-WindowsFeature` cmdlet, yang menginstal Web Server (IIS). Secara umum, `code` atribut dapat memiliki sejumlah baris, sehingga Anda dapat menjalankan cmdlet sebanyak yang Anda butuhkan.
- `not-if`— Atribut penjaga yang memastikan bahwa resep menginstal IIS hanya jika belum diinstal.

  Anda biasanya ingin resep menjadi idempoten, sehingga mereka tidak membuang waktu melakukan tugas yang sama lebih dari sekali.

Setiap sumber daya memiliki tindakan, yang menentukan tindakan yang harus diambil penyedia. Tidak ada tindakan eksplisit untuk contoh ini, sehingga penyedia mengambil `:run` tindakan default, yang menjalankan PowerShell skrip yang ditentukan. Untuk informasi selengkapnya, lihat Menjalankan PowerShell Skrip Windows.

layanan

A `service`mengelola layanan, layanan Web Server IIS (W3SVC) dalam kasus ini. Contoh menggunakan atribut default dan menentukan dua tindakan, `:start` dan`:enable`, yang memulai dan mengaktifkan IIS.

> **Note**
>
> Jika Anda ingin menginstal perangkat lunak yang menggunakan penginstal paket, seperti MSI, Anda dapat menggunakan sumber daya. `windows_package` Untuk informasi selengkapnya, lihat Menginstal Package.

Aktifkan Buku Masak Kustom

AWS OpsWorks Stacks menjalankan resep dari cache lokal pada setiap instance. Untuk menjalankan resep khusus Anda, Anda harus melakukan hal berikut:

- Simpan buku masak di repositori jarak jauh.

  AWS OpsWorks Stacks mengunduh buku masak dari repositori ini ke cache lokal masing-masing instance.
- Edit tumpukan untuk mengaktifkan buku masak khusus.

  Buku masak khusus dinonaktifkan secara default, jadi Anda harus mengaktifkan buku masak khusus untuk tumpukan dan memberikan URL repositori dan informasi terkait.

AWS OpsWorks Stacks mendukung arsip S3 dan repositori Git untuk buku masak kustom; contoh ini menggunakan arsip S3. Untuk informasi selengkapnya, lihat Repositori.

Untuk menggunakan arsip S3

1. Buat `.zip` arsip `iis-cookbook` direktori.

   AWS OpsWorks Stacks juga mendukung arsip `.tgz` (gzip compressed tar) untuk tumpukan Windows.
2. Unggah arsip ke bucket S3 di wilayah AS Barat (California Utara), dan buat file menjadi publik. Anda juga dapat menggunakan arsip S3 pribadi, tetapi arsip publik cukup untuk contoh ini dan agak lebih mudah untuk digunakan.

   a. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   b. Jika Anda belum memiliki bucket`us-west-1`, pilih Create Bucket dan buat bucket di wilayah AS Barat (California Utara).

   c. Dalam daftar bucket, pilih nama bucket yang ingin Anda unggah file, lalu pilih Unggah.

   d. Pilih Tambahkan File.

   e. Pilih file arsip yang akan diunggah, lalu pilih Buka.

   f. Di bagian bawah dialog Unggah - Pilih File dan Folder, pilih Atur Detail.

   g. Di bagian bawah dialog Set Detail, pilih Set Permissions.

   h. Dalam dialog Set Permissions, pilih Jadikan semuanya publik.

i. Di bagian bawah dialog Set Permissions, pilih Start Upload. Saat unggahan selesai, `iis-cookbook.zip` file akan muncul di bucket Anda.

j. Pilih bucket, lalu pilih tab Properties untuk bucket. Di samping Tautan, rekam URL file arsip untuk digunakan nanti.

Untuk informasi selengkapnya tentang mengunggah file ke bucket Amazon S3, lihat Bagaimana Cara Mengunggah File dan Folder ke Bucket S3? di Panduan Pengguna Konsol Amazon S3.

> **Important**
>
> Hingga saat ini, penelusuran hanya membebani Anda sedikit waktu; layanan AWS OpsWorks Stacks itu sendiri gratis. Namun, Anda harus membayar sumber daya AWS apa pun yang Anda gunakan, seperti penyimpanan Amazon S3. Segera setelah Anda mengunggah arsip, Anda mulai dikenakan biaya. Untuk informasi selengkapnya, lihat Harga AWS.

Untuk mengaktifkan buku masak khusus untuk tumpukan

1. Di konsol AWS OpsWorks Stacks, pilih Stack di panel navigasi, lalu pilih Stack Settings di kanan atas.
2. Di kanan atas halaman Pengaturan, pilih Edit.
3. Pada halaman Pengaturan, atur Gunakan buku masak Chef kustom ke Ya dan masukkan informasi berikut:
   - Jenis repositori - Arsip S3.
   - URL Repositori — URL S3 dari file arsip buku masak yang Anda rekam sebelumnya.
4. Pilih Simpan untuk memperbarui konfigurasi tumpukan.

AWS OpsWorks Stacks menginstal buku masak kustom Anda pada semua instance baru. Perhatikan bahwa AWS OpsWorks Stacks tidak secara otomatis menginstal atau memperbarui buku masak khusus pada instance online. Anda dapat melakukannya secara manual, seperti yang dijelaskan nanti.

Langkah 2.4: Tambahkan Layer IIS

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Buku masak Anda memiliki satu resep yang baru saja menginstal dan memulai IIS. Ini cukup untuk membuat layer dan memverifikasi bahwa Anda memiliki instance IIS yang berfungsi. Nanti, Anda akan menambahkan fungsionalitas penerapan aplikasi ke layer.

Buat Layer

Anda mulai dengan menambahkan lapisan ke tumpukan. Anda kemudian menambahkan fungsionalitas ke lapisan itu dengan menetapkan resep khusus ke peristiwa siklus hidup yang sesuai.

Untuk menambahkan lapisan IIS ke tumpukan

1. Pilih Layers di panel navigasi dan kemudian pilih Add a layer.
2. Konfigurasikan layer sebagai berikut:
   - Nama - **`IISExample`**
   - Nama pendek - **`iisexample`**

     AWS OpsWorks Stacks menggunakan nama pendek untuk mengidentifikasi layer secara internal. Anda juga menggunakan nama pendek untuk mengidentifikasi lapisan dalam resep, meskipun contoh ini tidak melakukannya. Anda dapat menentukan nama pendek apa pun, tetapi hanya dapat terdiri dari karakter alfanumerik huruf kecil dan sejumlah kecil tanda baca. Untuk informasi selengkapnya, lihat Lapisan Kustom.
3. Pilih Add Layer.

Jika Anda menambahkan instance IISWalkthrough pada saat ini dan memulainya, AWS OpsWorks Stacks akan secara otomatis menginstal buku masak tetapi tidak akan berjalan. `install.rb` Setelah sebuah instance online, Anda dapat menjalankan resep secara manual dengan

menggunakan perintah tumpukan Execute Recipes. Namun, pendekatan yang lebih baik adalah menetapkan resep ke salah satu peristiwa siklus hidup lapisan. AWS OpsWorks Stacks kemudian secara otomatis menjalankan resep pada titik yang sesuai dalam siklus hidup instance.

Instal dan mulai IIS segera setelah instance selesai booting. Untuk melakukan ini, tetapkan `install.rb` ke `Setup` acara layer.

Untuk menetapkan resep ke acara siklus hidup

1. Pilih Layers di panel navigasi
2. Di kotak untuk IISExamplelayer, pilih Resep.
3. Di kanan atas, pilih Edit.
4. Di bawah Resep Koki Kustom, di kotak Setup recipes, ketik**`iis-cookbook::install`**.

   > **Note**
   >
   > Gunakan *`cookbook-name`*`::`*`recipe-name`* untuk mengidentifikasi resep, di mana Anda menghilangkan `.rb` akhiran nama resep.

5. Pilih + untuk menambahkan resep ke layer. X merah muncul di sebelah resep untuk membuatnya mudah dihapus nanti.
6. Pilih Simpan untuk menyimpan konfigurasi baru. Resep Pengaturan kustom sekarang harus disertakan`iis-cookbook::install`.

Tambahkan Instance ke Layer dan Mulai

Anda dapat mencoba resepnya dengan menambahkan instance ke layer dan memulai instance. AWS OpsWorks Stacks secara otomatis menginstal buku masak dan berjalan `install.rb` selama penyiapan, segera setelah instance selesai booting.

Untuk menambahkan instance ke layer dan memulainya

1. Di panel navigasi AWS OpsWorks Stacks, pilih Instans.
2. Di bawah IISExamplelayer, pilih Add an instance.
3. Pilih ukuran yang sesuai. t2.micro (atau ukuran terkecil yang tersedia untuk Anda) harus cukup untuk contoh.

4. Pilih Tambahkan Instance. Secara default, AWS OpsWorks Stacks menghasilkan nama instance dengan menambahkan integer ke nama pendek layer, sehingga instance harus diberi nama iisexample1.
5. Pilih start di kolom Actions instance untuk memulai instance. AWS OpsWorks Stacks kemudian akan meluncurkan sebuah EC2 instance dan menjalankan resep Setup untuk mengkonfigurasinya. Jika layer memiliki resep Deploy pada saat ini, AWS OpsWorks Stacks akan menjalankannya setelah resep Pengaturan selesai.

   Prosesnya mungkin memakan waktu beberapa menit, di mana kolom Status menampilkan serangkaian status status. Saat Anda masuk ke status online, proses penyiapan selesai dan instance siap digunakan.

Konfirmasikan bahwa IIS Diinstal dan Dijalankan

Anda dapat menggunakan RDP untuk terhubung ke instans dan memverifikasi bahwa resep Pengaturan Anda bekerja dengan benar.

Untuk memverifikasi bahwa IIS diinstal dan berjalan

1. Pilih Instance di panel navigasi dan pilih rdp di kolom Actions instance iisexample1. AWS OpsWorks Stacks secara otomatis menghasilkan kata sandi RDP untuk Anda yang kedaluwarsa setelah jangka waktu tertentu.
2. Setel Sesi berlaku hingga 2 jam dan pilih Hasilkan Kata Sandi.
3. AWS OpsWorks Tumpukan menampilkan kata sandi dan juga, untuk kenyamanan Anda, nama DNS publik instans dan nama pengguna. Salin ketiganya dan klik Akui dan tutup.
4. Buka klien RDP Anda dan gunakan data dari Langkah 3 untuk terhubung ke instance.
5. Pada contoh, buka Windows Explorer dan periksa `C:` drive. Itu harus memiliki `C:\inetpub` direktori, yang dibuat oleh instalasi IIS.
6. Buka aplikasi Control Panel Administrative Tools, lalu buka Services. Anda akan melihat layanan IIS di dekat bagian bawah daftar. Ini bernama World Wide Web Publishing Service, dan statusnya harus berjalan.
7. Kembali ke konsol AWS OpsWorks Stacks dan pilih alamat IP publik instans iisexample1. Pastikan Anda melakukan ini di AWS OpsWorks Stacks, dan bukan di EC2 konsol Amazon. Ini secara otomatis mengirimkan permintaan HTTP ke alamat, yang seharusnya membuka halaman Selamat Datang IIS default.

Topik berikutnya membahas cara menerapkan aplikasi ke instance, halaman HTML statis sederhana untuk contoh ini. Namun, jika Anda ingin beristirahat, pilih stop di kolom Actions instans iisexample1 untuk menghentikan instance dan menghindari biaya yang tidak perlu. Anda dapat memulai ulang instance ketika Anda siap untuk melanjutkan.

Langkah 2.5: Menerapkan Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Instalasi IIS membuat `C:\inetpub\wwwroot` direktori untuk kode aplikasi Anda dan file terkait. Langkah selanjutnya adalah menginstal aplikasi di direktori itu. Untuk contoh ini, Anda akan menginstal halaman beranda HTML statis,`default.html`, di`C:\inetpub\wwwroot`. Anda dapat dengan mudah memperluas pendekatan umum untuk menangani skenario yang lebih kompleks, seperti aplikasi ASP.NET.

Anda dapat memasukkan file aplikasi di buku masak Anda dan `install.rb` menyalinnya. `C:\inetpub\wwwroot` Untuk contoh cara melakukannya, lihat Contoh 6: Membuat File. Namun, pendekatan ini tidak terlalu fleksibel atau efisien, dan biasanya lebih baik untuk memisahkan pengembangan buku masak dari pengembangan aplikasi.

Solusi yang lebih disukai adalah menerapkan resep penyebaran terpisah yang mengambil kode aplikasi dan file terkait dari repositori — repositori apa pun yang Anda inginkan, bukan hanya repositori buku masak — dan menginstalnya di setiap instance server IIS. Pendekatan ini memisahkan pengembangan buku masak dari pengembangan aplikasi dan, ketika Anda perlu memperbarui aplikasi Anda, ini memungkinkan Anda untuk menjalankan resep penerapan lagi tanpa harus memperbarui buku masak Anda.

Topik ini menunjukkan cara menerapkan resep penyebaran sederhana yang disebarkan `default.htm` ke server IIS Anda. Anda dapat dengan mudah memperluas contoh ini ke aplikasi yang lebih kompleks.

Topik

- Buat Aplikasi dan Simpan di Repositori
- Menerapkan Resep untuk Menyebarkan Aplikasi
- Perbarui Buku Masak Instance
- Tambahkan Resep ke Layer IIS Kustom
- Menambahkan Aplikasi
- Terapkan Aplikasi dan Buka Aplikasi

Buat Aplikasi dan Simpan di Repositori

Anda dapat menggunakan repositori apa pun yang Anda inginkan untuk aplikasi Anda. Untuk kesederhanaan, contoh ini disimpan `default.htm` di bucket S3 publik.

Untuk membuat aplikasi

1. Buat direktori bernama `iis-application` di lokasi yang nyaman di workstation Anda.
2. Tambahkan `default.htm` file `iis-application` dengan konten berikut.

```
<!DOCTYPE html>
<html>
  <head>
    <title>IIS Example</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

3. Buat bucket S3, unggah `default.htm` ke bucket, dan rekam URL untuk digunakan nanti. Untuk kesederhanaan, buat file publik.

> **Note**
>
> Ini adalah aplikasi yang sangat sederhana, tetapi Anda dapat memperluas prinsip-prinsip dasar untuk menangani aplikasi tingkat produksi.
>
> - Untuk aplikasi yang lebih kompleks dengan banyak file, biasanya lebih mudah untuk membuat arsip.zip `iis-application` dan mengunggahnya ke bucket S3 Anda.

Anda kemudian dapat mengunduh file.zip dan mengekstrak konten ke direktori yang sesuai. Tidak perlu mengunduh banyak file, membuat struktur direktori, atau sebagainya.

- Untuk aplikasi produksi, Anda mungkin ingin menyimpan file Anda secara pribadi. Untuk contoh cara mengunduh file resep dari bucket S3 pribadi, lihatMenggunakan SDK for Ruby AWS OpsWorks pada Instans Windows Stacks.
- Anda dapat menyimpan aplikasi Anda di repositori yang sesuai.

  Anda biasanya mengunduh aplikasi dengan menggunakan API publik repositori. Contoh ini menggunakan Amazon S3 API. Jika, misalnya, Anda menyimpan aplikasi Anda GitHub, Anda dapat menggunakan GitHub API.

Menerapkan Resep untuk Menyebarkan Aplikasi

Tambahkan resep bernama `deploy.rb` ke `iis-cookbook` `recipes` direktori, dengan konten berikut.

```
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
```

Contoh ini menggunakan SDK for Ruby v2 untuk mengunduh file. Namun, AWS OpsWorks Stacks tidak menginstal SDK ini pada instance Windows, jadi resep dimulai dengan `chef_gem`sumber daya, yang menangani tugas itu.

> **Note**
>
> `chef_gem`Sumber daya menginstal permata ke dalam versi Ruby khusus Chef, yang merupakan versi yang digunakan resep. Jika Anda ingin menginstal permata untuk versi Ruby seluruh sistem, gunakan sumber daya gem_package.

Sebagian besar resep adalah `ruby_block`sumber daya, yang menjalankan blok kode Ruby yang menggunakan SDK for Ruby untuk diunduh. `default.htm` Kode dalam `ruby_block` dapat dibagi menjadi beberapa bagian berikut, yang sesuai dengan komentar bernomor dalam contoh kode.

1: Tentukan Bundel Sertifikat

Amazon S3 menggunakan SSL, jadi Anda memerlukan sertifikat yang sesuai untuk mengunduh objek dari bucket S3. SDK for Ruby v2 tidak menyertakan bundel sertifikat, jadi Anda harus menyediakannya dan mengonfigurasi SDK for Ruby untuk menggunakannya. AWS OpsWorks Stacks tidak menginstal bundel sertifikat secara langsung, tetapi menginstal Git, yang mencakup bundel sertifikat (`curl-ca-bundle.crt`). Untuk kenyamanan, contoh ini mengonfigurasi SDK for Ruby untuk menggunakan bundel sertifikat Git untuk SSL. Anda juga dapat menginstal bundel Anda sendiri dan mengonfigurasi SDK yang sesuai.

2: Ambil Data Repositori

Untuk mengunduh objek dari Amazon S3, Anda memerlukan wilayah AWS, nama bucket, dan nama kunci. Seperti yang dijelaskan nanti, contoh ini memberikan informasi ini dengan mengaitkan satu set variabel lingkungan dengan aplikasi. Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks menambahkan satu set atribut ke objek node instance. Atribut ini pada dasarnya adalah tabel hash yang berisi konfigurasi aplikasi, termasuk variabel lingkungan. Atribut aplikasi untuk aplikasi ini akan terlihat seperti berikut, dalam 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
}
```

Variabel lingkungan aplikasi disimpan dalam `[:environment]` atribut. Untuk mengambilnya, gunakan kueri penelusuran Chef untuk mengambil tabel hash aplikasi, yang berada di bawah node. `aws_opsworks_app` Aplikasi ini akan didefinisikan sebagai `other` jenisnya, sehingga kueri mencari aplikasi jenis itu. Resepnya memanfaatkan fakta bahwa hanya ada satu aplikasi pada contoh ini, jadi tabel hash yang menarik adalah adil`app[0]`. Untuk kenyamanan, resep kemudian menetapkan wilayah, ember, dan nama file ke variabel.

Untuk informasi selengkapnya tentang cara menggunakan pencarian Chef, lihat. Memperoleh Nilai Atribut dengan Pencarian Chef

3: Unduh file

Bagian ketiga dari resep membuat objek klien S3 dan menggunakan `get_object` metodenya untuk mengunduh `default.htm` ke `C:\inetpub\wwwroot` direktori instance.

> **Note**
>
> Resep adalah aplikasi Ruby, jadi kode Ruby tidak harus ada di file. `ruby_block` Namun, kode di badan resep berjalan lebih dulu, diikuti oleh sumber daya, secara berurutan. Untuk contoh ini, jika Anda meletakkan kode unduhan di badan resep, itu akan gagal karena `chef_gem` sumber daya belum menginstal SDK for Ruby. Kode dalam `ruby_block` sumber daya dijalankan ketika sumber daya dijalankan, setelah sumber `chef_gem` daya menginstal SDK for Ruby.

Perbarui Buku Masak Instance

AWS OpsWorks Stacks secara otomatis menginstal buku masak khusus pada instance baru. Namun, Anda bekerja dengan instance yang ada, jadi Anda harus memperbarui buku masak Anda secara manual.

Untuk memperbarui buku masak instans

1. Buat `.zip` arsip`iis-cookbook`, dan unggah ke bucket S3.

   Ini menimpa buku masak yang ada, tetapi URL-nya tetap sama, jadi Anda tidak perlu memperbarui konfigurasi tumpukan.
2. Jika instans Anda tidak online, restart.
3. Setelah instance online, pilih Stack di panel navigasi, lalu pilih Run Command.
4. Untuk Perintah, pilih Perbarui Buku Masak Kustom. Perintah ini menginstal buku masak yang diperbarui pada instance.
5. Pilih Perbarui Buku Masak Kustom. Perintah mungkin memakan waktu beberapa menit untuk menyelesaikannya.

Tambahkan Resep ke Layer IIS Kustom

Seperti halnya`install.rb`, cara yang lebih disukai untuk menangani penerapan adalah dengan menetapkan `deploy.rb` ke peristiwa siklus hidup yang sesuai. Anda biasanya menetapkan resep penyebaran ke acara Deploy, dan mereka disebut secara kolektif sebagai resep Deploy. Menetapkan resep ke acara Deploy tidak memicu peristiwa. Sebagai gantinya:

- Untuk instance baru, AWS OpsWorks Stacks secara otomatis menjalankan resep Deploy setelah resep Setup selesai, sehingga instance baru secara otomatis memiliki versi aplikasi saat ini.
- Untuk instance online, Anda menggunakan perintah deploy untuk menginstal aplikasi baru atau yang diperbarui secara manual.

  Perintah ini memicu peristiwa Deploy pada instance tumpukan, yang menjalankan resep Deploy.

Untuk menetapkan deploy.rb ke event Deploy layer

1. Pilih Layers di panel navigasi, lalu pilih Recipes di bawah Layer IISExample.
2. Di bawah Custom Chef Recipes, tambahkan **`iis-cookbook::deploy`** ke kotak Deploy recipes dan pilih + untuk menambahkan resep ke layer.
3. Pilih Simpan untuk menyimpan konfigurasi baru. Resep Deploy khusus sekarang harus disertakan`iis-cookbook::deploy`.

Menambahkan Aplikasi

Tugas terakhir adalah menambahkan aplikasi ke tumpukan untuk mewakili aplikasi Anda di lingkungan AWS OpsWorks Stacks. Aplikasi menyertakan metadata seperti nama tampilan aplikasi, dan data yang diperlukan untuk mengunduh aplikasi dari repositorinya.

Untuk menambahkan aplikasi ke tumpukan

1. Pilih Aplikasi di panel navigasi, lalu pilih Tambahkan aplikasi.
2. Konfigurasikan aplikasi dengan pengaturan berikut.
   - Nama - I **`IIS-Example-App`**
   - Jenis Repositori — Lainnya
   - Variabel Lingkungan - Tambahkan tiga variabel lingkungan berikut:
     - **`S3REGION`**— Wilayah ember (dalam hal ini,`us-west-1`).

- **`BUCKET`**— Nama ember, seperti`windows-example-app`.
- **`FILENAME`**— Nama file:**`default.htm`**.

3. Terima nilai default untuk setelan yang tersisa, lalu pilih Tambah Aplikasi untuk menambahkan aplikasi ke tumpukan.

> **Note**
>
> Contoh ini menggunakan variabel lingkungan untuk menyediakan data unduhan. Pendekatan alternatif adalah dengan menggunakan jenis repositori Arsip S3 dan menyediakan URL file. AWS OpsWorks Stacks menambahkan informasi, bersama dengan data opsional, seperti kredensil AWS Anda, ke atribut aplikasi. `app_source` Resep penerapan Anda harus mendapatkan URL dari atribut aplikasi dan mengurainya untuk mengekstrak wilayah, nama bucket, dan nama file.

Terapkan Aplikasi dan Buka Aplikasi

AWS OpsWorks Stacks secara otomatis menyebarkan aplikasi ke instance baru, tetapi tidak ke instance online. Karena instance Anda sudah berjalan, Anda harus menerapkan aplikasi secara manual.

Untuk menyebarkan aplikasi

1. Pilih Aplikasi di panel navigasi, lalu pilih deploy di kolom Tindakan aplikasi.
2. Perintah harus diatur ke Deploy. Pilih Deploy di kanan bawah halaman Deploy App. Perintah mungkin memakan waktu beberapa menit untuk menyelesaikannya.

   Setelah penerapan selesai, Anda kembali ke halaman Aplikasi. Indikator Status menunjukkan keberhasilan dalam warna hijau, dan nama aplikasi memiliki tanda centang hijau di sebelahnya untuk menunjukkan penerapan yang berhasil.

> **Note**
>
> Aplikasi Windows selalu merupakan jenis aplikasi Lain, jadi menerapkan aplikasi melakukan hal berikut:

- Menambahkan data aplikasi ke konfigurasi stack dan atribut deployment, seperti yang dijelaskan sebelumnya.
- Memicu peristiwa Deploy pada instance tumpukan, yang menjalankan resep Deploy kustom Anda.

> **Note**
>
> Untuk informasi selengkapnya tentang cara memecahkan masalah penerapan atau aplikasi yang gagal, lihat. Resep Debugging

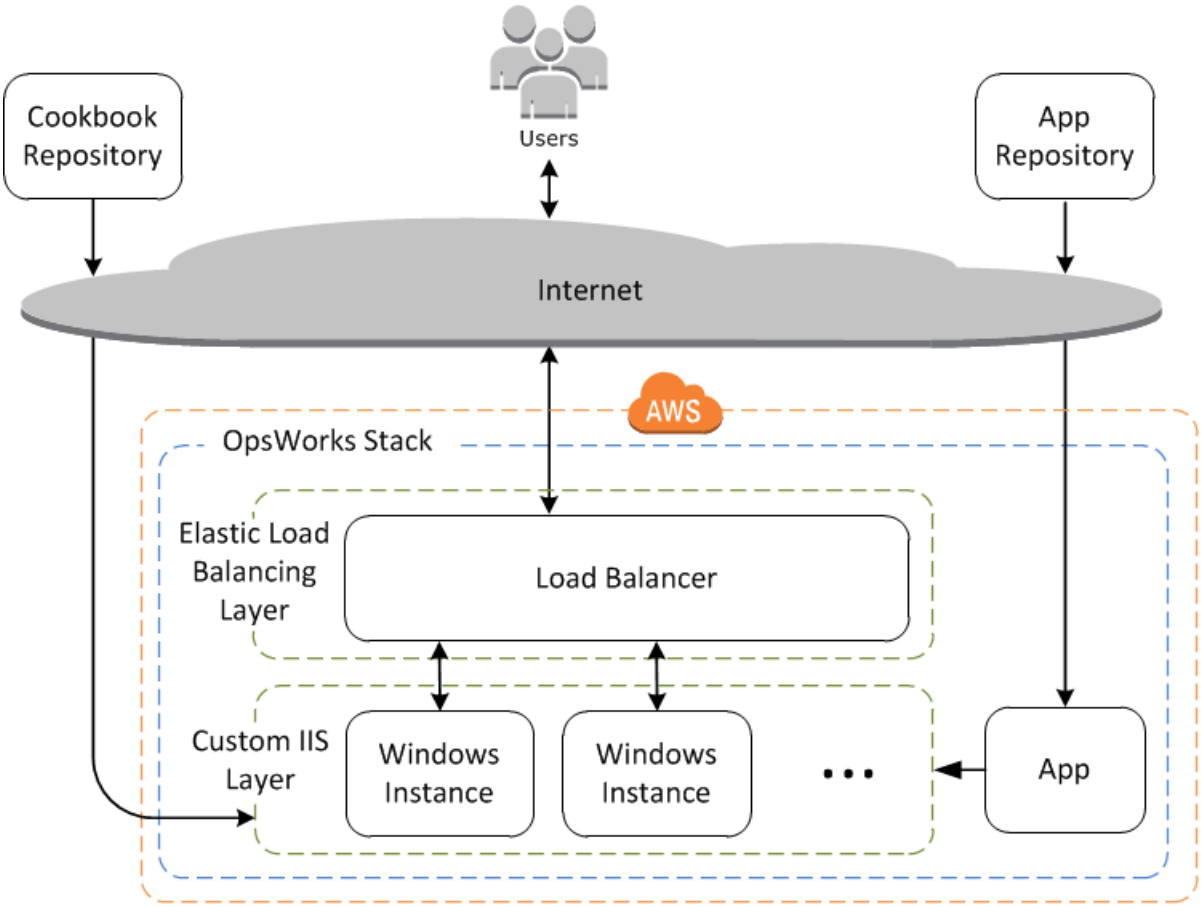
Aplikasi ini sekarang diinstal. Anda dapat membukanya dengan memilih Instans di panel Navigasi, lalu memilih alamat IP publik instans. Ini mengirimkan permintaan HTTP ke instance, dan Anda akan melihat sesuatu seperti berikut di browser Anda.



Langkah 3: Skala Keluar IISExample

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika permintaan pengguna masuk Anda mulai mendekati batas apa yang dapat Anda tangani dengan satu instans t2.micro, Anda perlu meningkatkan kapasitas server Anda. Anda dapat pindah ke contoh yang lebih besar, tetapi itu memiliki batas. Pendekatan yang lebih fleksibel adalah menambahkan instance ke tumpukan Anda, dan menempatkannya di belakang penyeimbang beban. Arsitektur dasar terlihat seperti berikut ini.



Di antara keuntungan lainnya, pendekatan ini jauh lebih kuat daripada satu contoh besar.

- Jika salah satu instans Anda gagal, penyeimbang beban akan mendistribusikan permintaan masuk ke instans yang tersisa, dan aplikasi Anda akan terus berfungsi.
- Jika Anda menempatkan instance di Availability Zone yang berbeda (praktik yang disarankan), aplikasi Anda akan terus berfungsi meskipun Availability Zone mengalami masalah.

AWS OpsWorks Tumpukan membuatnya mudah untuk skala tumpukan. Bagian ini menjelaskan dasar-dasar cara menskalakan tumpukan dengan menambahkan instance Server Aplikasi PHP 24/7 kedua ke IISExample dan menempatkan kedua instance di belakang penyeimbang beban Elastic Load Balancing. Anda dapat dengan mudah memperpanjang prosedur untuk menambahkan jumlah instance 24/7 yang sewenang-wenang, atau Anda dapat menggunakan instance berbasis waktu agar AWS OpsWorks Stacks menskalakan tumpukan Anda secara otomatis. Untuk informasi selengkapnya, lihat Mengelola beban dengan instans berbasis waktu dan berbasis beban.

Tambahkan Load Balancer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Elastic Load Balancing adalah layanan AWS yang secara otomatis mendistribusikan lalu lintas aplikasi masuk di beberapa instans Amazon. EC2 Penyeimbang beban dapat melayani dua tujuan. Yang jelas adalah menyamakan beban pada server aplikasi Anda. Banyak situs lebih suka mengisolasi server aplikasi dan database mereka dari akses pengguna langsung. Selain mendistribusikan lalu lintas, Elastic Load Balancing melakukan hal berikut:

- Mendeteksi EC2 contoh Amazon yang tidak sehat.

  Ini mengalihkan lalu lintas ke contoh sehat yang tersisa sampai kasus yang tidak sehat dipulihkan.
- Secara otomatis menskalakan kapasitas penanganan permintaan sebagai respons terhadap lalu lintas yang masuk.

> **Note**
>
> AWS OpsWorks Stacks tidak mendukung Application Load Balancer. Anda hanya dapat menggunakan Classic Load Balancer dengan AWS OpsWorks Stacks.

Meskipun Elastic Load Balancing sering disebut sebagai lapisan, ia bekerja sedikit berbeda dari lapisan bawaan lainnya. Alih-alih membuat layer dan menambahkan instance ke dalamnya, Anda membuat penyeimbang beban Elastic Load Balancing dengan menggunakan konsol EC2 Amazon dan kemudian melampirkannya ke salah satu layer yang ada, biasanya lapisan server aplikasi. AWS OpsWorks Stacks kemudian mendaftarkan instance layer yang ada dengan layanan dan secara otomatis menambahkan instance baru. Prosedur berikut menjelaskan cara menambahkan penyeimbang beban.

Untuk melampirkan penyeimbang beban ke lapisan IIS kustom

1. Gunakan EC2 konsol Amazon untuk membuat penyeimbang beban baru. IISExample Untuk informasi selengkapnya, lihat Memulai Elastic Load Balancing. Saat Anda menjalankan wizard Create Load Balancer, konfigurasikan penyeimbang beban sebagai berikut:

   1: Tentukan Load Balancer

   Tetapkan penyeimbang beban nama yang mudah dikenali, seperti IIS-LB, untuk membuatnya lebih mudah ditemukan di konsol Stacks. AWS OpsWorks Terima default untuk pengaturan yang tersisa, lalu pilih Berikutnya: Tetapkan Grup Keamanan.

   2: Tetapkan Grup Keamanan

   Jika akun Anda mendukung VPC default, wizard akan menampilkan halaman ini untuk menentukan grup keamanan penyeimbang beban. Itu tidak menampilkan halaman ini untuk EC2 Klasik.

   Untuk panduan ini, tentukan grup keamanan VPC default, lalu pilih Berikutnya: Konfigurasi Pengaturan Keamanan.

   3: Konfigurasikan Pengaturan Keamanan

   Panduan ini memang mengharuskan penyeimbang beban Anda untuk menggunakan pendengar yang aman (yaitu, HTTPS atau SSL pada koneksi front-end-nya), jadi pilih Berikutnya: Konfigurasi Pemeriksaan Kesehatan untuk melanjutkan.

   4: Konfigurasikan Pemeriksaan Kesehatan

   Atur jalur ping ke**/**. Terima default untuk pengaturan yang tersisa, lalu pilih Berikutnya: Tambahkan Instans. EC2

   5: Tambahkan EC2 Contoh

   AWS OpsWorks Stacks secara otomatis menangani pendaftaran instance dengan penyeimbang beban. Pilih Berikutnya Tambahkan Tag untuk melanjutkan.

   6: Tambahkan Tag

   Anda tidak akan menggunakan tag untuk contoh ini. Pilih Tinjau dan Buat.

   7: Ulasan

   Tinjau pilihan Anda dan pilih Buat dan kemudian Tutup, yang meluncurkan penyeimbang beban.

2. Jika akun Anda mendukung VPC default, setelah meluncurkan penyeimbang beban, Anda harus memastikan bahwa grup keamanannya memiliki aturan masuk yang sesuai. Aturan default tidak menerima lalu lintas masuk apa pun.

    1. Pilih Grup Keamanan di panel EC2 navigasi Amazon.
    2. Pilih grup keamanan VPC default
    3. Pada tab Ke dalam, pilih Edit.
    4. Untuk panduan ini, atur Source to Anywhere, yang mengarahkan penyeimbang beban untuk menerima lalu lintas masuk dari alamat IP apa pun.
    5. Klik Simpan.
3. Kembali ke konsol AWS OpsWorks Stacks. Pada halaman Layers, pilih Network.
4. Di bawah Elastic Load Balancing, pilih load balancer IIS-LB yang Anda buat di Langkah 1, lalu klik Simpan.

    Setelah Anda melampirkan load balancer ke layer, AWS OpsWorks Stacks secara otomatis mendaftarkan instance layer saat ini dan menambahkan instance baru saat mereka online.
5. Pada halaman Layers, klik nama load balancer untuk membuka halaman detailnya. Pemeriksaan hijau di sebelah instance pada halaman penyeimbang beban menunjukkan bahwa instance telah lulus pemeriksaan kesehatan.

Anda sekarang dapat menjalankan IIS-Example-App dengan mengirimkan permintaan ke penyeimbang beban.

Untuk IIS-Example-App menjalankan penyeimbang beban

1. Pilih Layers. Penyeimbang beban IIS-ELB harus terdaftar sebagai lapisan dan kolom Health harus memiliki satu instance berwarna hijau, yang menunjukkan contoh yang sehat.
2. Pilih nama DNS load balancer untuk dijalankan. IIS-Example-App Itu harus terdaftar di bawah nama penyeimbang beban dan terlihat sepertiIIS-LB-1802910859.us-west-2.elb.amazonaws.com. Penyeimbang beban meneruskan permintaan ke instance dan mengembalikan respons, yang seharusnya terlihat persis sama dengan respons yang Anda dapatkan saat Anda mengklik alamat IP publik instans.

Anda hanya memiliki satu contoh pada saat ini, jadi penyeimbang beban tidak benar-benar menambahkan banyak. Namun, Anda sekarang dapat menambahkan instance tambahan ke layer.

Untuk menambahkan instance ke layer

1. Pilih Instances dan kemudian + instance untuk menambahkan instance lain ke layer.
2. Mulai instans.

Karena ini adalah instance baru, AWS OpsWorks Stacks secara otomatis menginstal buku masak kustom saat ini dan menerapkan versi aplikasi saat ini selama penyiapan. Ketika instance online, AWS OpsWorks Stacks secara otomatis menambahkannya ke penyeimbang beban, sehingga instans Anda akan segera mulai menangani permintaan. Untuk memverifikasi bahwa aplikasi masih berfungsi, Anda dapat memilih nama DNS penyeimbang beban lagi.

Langkah Berikutnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Panduan ini membawa Anda melalui dasar-dasar pengaturan tumpukan server aplikasi Windows sederhana. Berikut adalah beberapa saran untuk apa yang harus dilakukan selanjutnya.

- Jika Anda ingin tahu lebih banyak, Memulai: Buku Masak berikan pengantar tutorial untuk mengimplementasikan buku masak, dan sertakan sejumlah contoh khusus AWS OpsWorks Stacks.
- Anda dapat menambahkan lapisan Amazon Relational Database Service (Amazon RDS) ke tumpukan untuk digunakan sebagai server database backend. Untuk informasi tentang cara menghubungkan aplikasi Anda ke database, lihatMenggunakan Resep Kustom.

Memulai dengan Buku Masak di Tumpukan AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan AWS OpsWorks Stacks tingkat produksi biasanya memerlukan beberapa penyesuaian, yang sering berarti menerapkan buku masak Chef khusus. Buku masak adalah file paket yang berisi informasi konfigurasi, termasuk instruksi yang disebut resep. Resep adalah seperangkat satu atau lebih instruksi, ditulis dengan sintaks bahasa Ruby, yang menentukan sumber daya yang akan digunakan dan urutan penerapan sumber daya tersebut. Sumber daya, seperti yang digunakan dalam Chef, adalah pernyataan kebijakan konfigurasi. Panduan ini memberikan pengantar dasar untuk mengimplementasikan buku masak Chef untuk Stacks. AWS OpsWorks Untuk mempelajari selengkapnya tentang Chef, buku masak, resep, dan sumber daya, lihat tautan diLangkah Berikutnya.

Panduan ini sebagian besar menjelaskan cara membuat buku masak Anda sendiri. Anda juga dapat menggunakan buku masak yang disediakan komunitas yang tersedia di situs web seperti Chef Supermarket. Untuk membantu Anda memulai dengan buku masak komunitas, kami menyertakan instruksi untuk menggunakan buku masak komunitas dari Chef Supermarket nanti dalam penelusuran.

Sebelum memulai panduan ini, selesaikan beberapa langkah pengaturan. Jika Anda telah menyelesaikan salah satu panduan lain di chapter ini, sepertiMemulai: Contoh, maka Anda telah memenuhi prasyarat untuk panduan ini dan dapat melompat untuk memulai panduan ini. Jika tidak, pastikan untuk menyelesaikan prasyarat, dan kemudian kembali ke panduan ini.

Topik

- Langkah 1: Buat Buku Masak
- Langkah 2: Buat Stack dan Komponennya
- Langkah 3: Jalankan dan Uji Resepnya
- Langkah 4: Memperbarui Buku Masak untuk Instal Package

- Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya
- Langkah 6: Perbarui Buku Masak untuk Tambahkan Pengguna
- Langkah 7: Perbarui Buku Masak untuk Membuat Direktori
- Langkah 8: Perbarui Buku Masak untuk Membuat dan Menyalin File
- Langkah 9: Perbarui Buku Masak untuk Menjalankan Perintah
- Langkah 10: Perbarui Cookbook untuk Menjalankan Script
- Langkah 11: Perbarui Buku Masak untuk Mengelola Layanan
- Langkah 12: Perbarui Buku Masak untuk Menggunakan JSON Kustom
- Langkah 13: Perbarui Buku Masak untuk Menggunakan Tas Data
- Langkah 14: Perbarui Cookbook untuk Menggunakan Iterasi
- Langkah 15: Perbarui Buku Masak untuk Menggunakan Logika Bersyarat
- Langkah 16: Perbarui Buku Masak untuk Menggunakan Buku Masak Komunitas
- Langkah 17: (Opsional) Hapus
- Langkah Berikutnya

Langkah 1: Buat Buku Masak

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Mulailah dengan membuat buku masak. Buku masak ini tidak akan banyak membantu untuk memulai, tetapi berfungsi sebagai dasar untuk sisa panduan ini.

> **Note**
>
> Langkah ini menunjukkan cara membuat buku masak secara manual. Anda dapat membuat buku masak dalam waktu yang lebih singkat dengan kit pengembangan Chef (Chef DK)

dengan menjalankan perintah chef generate cookbookdi workstation lokal Anda. Namun, perintah ini membuat beberapa folder dan file yang tidak Anda perlukan untuk panduan ini.

Untuk membuat buku masak

1. Di workstation lokal Anda, buat direktori bernamaopsworks_cookbook_demo. Anda dapat menggunakan nama yang berbeda, tetapi pastikan untuk menggantinya opsworks_cookbook_demo di seluruh panduan ini.
2. Di opsworks_cookbook_demo direktori, buat file bernama metadata.rb menggunakan editor teks. Tambahkan kode berikut untuk menentukan nama buku masak. Untuk informasi lebih lanjut tentangmetadata.rb, lihat metadata.rb di situs web Chef.

   ```
   name "opsworks_cookbook_demo"
   ```
3. Di opsworks_cookbook_demo direktori, buat subdirektori bernamarecipes. Subdirektori ini berisi semua resep yang Anda buat untuk buku masak panduan ini.
4. Di direktori recipes, buat file baru bernama default.rb. File ini berisi resep dengan nama yang sama dengan file, tetapi tanpa ekstensi file:default. Tambahkan satu baris kode berikut ke default.rb file. Kode ini adalah resep satu baris yang menampilkan pesan sederhana di log saat resep berjalan:

   ```
   Chef::Log.info("********** Hello, World! **********")
   ```
5. Di terminal atau command prompt, gunakan tar perintah untuk membuat file bernamaopsworks_cookbook_demo.tar.gz, yang berisi opsworks_cookbook_demo direktori dan isinya. Misalnya:

   ```
   tar -czvf opsworks_cookbook_demo.tar.gz opsworks_cookbook_demo/
   ```

   Anda dapat menggunakan nama file yang berbeda, tetapi pastikan untuk menggantinya opsworks_cookbook_demo.tar.gz di seluruh panduan ini.

   > **Note**
   >
   > Saat Anda membuat tar file di Windows, direktori teratas harus menjadi direktori induk buku masak. Panduan ini telah diuji di Linux dengan tar perintah yang disediakan oleh tar paket dan pada Windows dengan tar perintah yang disediakan oleh Git Bash.

Menggunakan perintah atau program lain untuk membuat file TAR terkompresi (.tar.gz) mungkin tidak berfungsi seperti yang diharapkan.

6. Buat bucket S3, atau gunakan bucket yang ada. Untuk informasi selengkapnya, lihat Membuat Bucket.
7. Unggah `opsworks_cookbook_demo.tar.gz` file ke bucket S3. Untuk informasi selengkapnya, lihat Menambahkan Objek ke Bucket.

Anda sekarang memiliki buku masak yang akan Anda gunakan selama panduan ini.

Pada langkah berikutnya, Anda membuat tumpukan AWS OpsWorks Stacks yang akan Anda gunakan nanti untuk mengunggah buku masak Anda dan menjalankan resep buku masak.

Langkah 2: Buat Stack dan Komponennya

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Buat tumpukan AWS OpsWorks Stacks dan komponennya, yang mencakup lapisan dan instance. Pada langkah selanjutnya, Anda mengunggah buku masak Anda ke instance dan kemudian menjalankan resep buku masak pada contoh itu.

Untuk membuat tumpukan

1. Masuk ke konsol AWS OpsWorks Stacks di https://console.aws.amazon.com/opsworks.
2. Lakukan salah satu langkah berikut, jika diterapkan:

   - Jika halaman Selamat Datang di AWS OpsWorks Tumpukan ditampilkan, pilih Tambahkan tumpukan pertama Anda atau Tambahkan tumpukan AWS OpsWorks tumpukan pertama Anda (kedua pilihan melakukan hal yang sama). Halaman Tambahkan tumpukan ditampilkan.
   - Jika halaman OpsWorks Dasbor ditampilkan, pilih Tambah tumpukan. Halaman Tambahkan Tumpukan ditampilkan.

3. Pilih tumpukan Chef 12.
4. Di kotak nama Stack, ketikkan nama tumpukan, misalnya**`MyCookbooksDemoStack`**. Anda dapat mengetikkan nama yang berbeda, tetapi pastikan untuk menggantinya `MyCookbooksDemoStack` di seluruh panduan ini.
5. Untuk Wilayah, pilih AS Barat (Oregon).
6. Untuk VPC, lakukan salah satu langkah berikut:
   - Jika VPC tersedia, pilihlah. Untuk informasi selengkapnya, lihat Menjalankan Stack di VPC.
   - Jika tidak, pilih Tanpa VPC.
7. Untuk Gunakan buku masak Chef khusus, pilih Ya.
8. Untuk jenis Repositori, pilih Arsip S3.

   > **Note**
   >
   > Dalam Memulai: Linux panduan, Anda memilih Http Archive. Pastikan untuk memilih Arsip S3 di sini sebagai gantinya.

9. Untuk URL Repositori, ketik jalur ke `opsworks_cookbook_demo.tar.gz` file Anda di S3. Untuk mendapatkan jalur, di konsol S3, pilih `opsworks_cookbook_demo.tar.gz` file. Pada panel Properties, salin nilai bidang Link. (Itu harus serupa dengan ini:`https://s3.amazonaws.com/amzn-s3-demo-bucket/opsworks_cookbook_demo.tar.gz`.)
10. Jika bucket S3 Anda bersifat pribadi, yang merupakan default, maka untuk Access key ID dan Secret access key, ketikkan ID kunci akses dan kunci akses rahasia pengguna IAM yang Anda gunakan untuk panduan ini. Untuk informasi selengkapnya, lihat Mengedit Izin Objek dan Berbagi Objek dengan Orang Lain.
11. Biarkan default untuk hal berikut ini:
    - Zona Ketersediaan default (us-west-2a)
    - Sistem operasi default (Linux dan Amazon Linux 2016.09)
    - Kunci SSH default (Jangan gunakan kunci SSH default)
    - Warna tumpukan (biru tua)
12. Pilih Lanjutan.
13. Untuk IAM role, lakukan salah satu hal berikut ini:
    - Jika aws-opsworks-service-roletersedia, pilihlah.

- Jika tidak aws-opsworks-service-roletersedia, pilih Peran IAM Baru.

14. Untuk profil Instans IAM default, lakukan salah satu hal berikut ini:
    - Jika aws-opsworks-ec2 peran tersedia, pilihlah.
    - Jika aws-opsworks-ec2 peran tidak tersedia, pilih Profil instans IAM baru.
15. Biarkan default untuk hal berikut ini:
    - Jenis perangkat root default (didukung EBS)
    - Tema nama host (Layer Dependent)
    - OpsWorks Versi agen (versi terbaru)
    - Custom Chef JSON (kosong)
    - Keamanan, Gunakan grup OpsWorks keamanan (Ya)
16. Pilih Tambahkan tumpukan. AWS OpsWorks Stacks membuat tumpukan dan menampilkan MyCookbooksDemoStackhalaman.

Untuk membuat lapisan

1. Pada panel navigasi, pilih Layers. Halaman Layers ditampilkan.
2. Pilih Tambahkan lapisan.
3. Pada OpsWorkstab, untuk Nama, ketik**`MyCookbooksDemoLayer`**. Anda dapat mengetikkan nama yang berbeda, tetapi pastikan untuk menggantinya `MyCookbooksDemoLayer` di seluruh panduan ini.
4. Untuk nama pendek, ketik**`cookbooks-demo`**. Anda dapat mengetikkan nama yang berbeda, tetapi pastikan untuk menggantinya `cookbooks-demo` di seluruh panduan ini.
5. Pilih Tambahkan lapisan. AWS OpsWorks Stacks menambahkan layer dan menampilkan halaman Layers.

Untuk membuat dan memulai instans

1. Pada panel navigasi, pilih Instans. Halaman Instans ditampilkan.
2. Pilih Tambahkan instance.
3. Pada tab Baru, pilih Advanced.
4. Biarkan default untuk hal berikut ini:

- Nama host (buku masak-demo1)
- Ukuran (c3.large)
- Subnet (*IP address*us-barat-2a)
- Jenis penskalaan (24/7)
- Kunci SSH (Jangan gunakan kunci SSH default)
- Sistem operasi (Amazon Linux 2016.09)
- OpsWorks Versi agen (Mewarisi dari tumpukan)
- Penyewaan (Default - Mengandalkan pengaturan VPC)
- Jenis perangkat root (didukung EBS)
- Jenis volume (Tujuan Umum (SSD))
- Ukuran volume (8)

5. Pilih Tambahkan contoh.
6. Untuk MyCookbooksDemoLayer, untuk buku masak-demo1, untuk Tindakan, pilih mulai. Jangan lanjutkan sampai Status berubah menjadi online. Proses ini mungkin memakan waktu beberapa menit, jadi bersabarlah.

Anda sekarang memiliki tumpukan, lapisan, dan instance yang buku masak disalin secara otomatis dari ember S3 Anda. Pada langkah berikutnya, Anda akan menjalankan dan menguji resep default dari dalam buku masak pada instance.

Langkah 3: Jalankan dan Uji Resepnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jalankan dan uji `default` resep dari dalam buku masak yang disalin AWS OpsWorks Stacks ke instance. Seperti yang akan Anda ingat, ini adalah resep satu baris yang menampilkan pesan sederhana di log saat resep berjalan.

Untuk menjalankan resep

1. Pada panel navigasi, pilih Stack. Halaman MyCookbooksDemoStack ditampilkan.
2. Pilih Run Command. Halaman Run Command ditampilkan.
3. Untuk Command, pilih Execute Recipes.
4. Agar Resep dapat dijalankan, ketik**`opsworks_cookbook_demo::default`**.

   `opsworks_cookbook_demo`adalah nama buku masak seperti yang didefinisikan dalam `metadata.rb` file. **`default`**adalah nama resep yang akan dijalankan, yaitu nama `default.rb` file di `recipes` subdirektori buku masak, tanpa ekstensi file.
5. Tinggalkan pengaturan default berikut:

   - Komentar (kosong)
   - Advanced, Custom Chef JSON (kosong)
   - Contoh (Pilih semua dicentang, dicentang, buku masak-demo1 MyCookbooksDemoLayerdicentang)
6. Pilih Jalankan Resep. Halaman perintah Running execute_recipes ditampilkan. Jangan lanjutkan sampai Status berubah menjadi berhasil. Proses ini mungkin memakan waktu beberapa menit, jadi bersabarlah.

Untuk memeriksa hasil resep

1. Dengan halaman perintah Running execute_recipes ditampilkan, untuk cookbooks-demo1, untuk Log, pilih tampilkan. Halaman log execute_recipes ditampilkan.
2. Gulir ke bawah log dan temukan entri yang terlihat serupa dengan yang berikut ini:

```
[2015-11-13T19:14:39+00:00] INFO: ********** Hello, World! **********
```

Anda telah berhasil menjalankan resep pertama Anda! Pada langkah berikutnya, Anda akan memperbarui buku masak Anda dengan menambahkan resep yang menginstal paket pada instance.

Langkah 4: Memperbarui Buku Masak untuk Instal Package

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang menginstal pada instance paket yang berisi editor teks populer GNU Emacs.

Meskipun Anda dapat dengan mudah masuk ke instance dan menginstal paket sekali, menulis resep memungkinkan Anda menjalankan resep dari AWS OpsWorks Stacks sekali untuk menginstal beberapa paket pada beberapa instance dalam tumpukan secara bersamaan.

Untuk memperbarui buku masak untuk menginstal paket

1. Kembali ke workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `install_package.rb` dengan kode berikut:

   ```
   package "Install Emacs" do
     package_name "emacs"
   end
   ```

   Resep ini menginstal `emacs` paket pada instance. (Untuk informasi lebih lanjut, buka paket.)

   > **Note**
   >
   > Anda dapat memberikan resep nama file apa pun yang Anda inginkan. Pastikan untuk menentukan nama resep yang benar kapan pun Anda ingin AWS OpsWorks Stacks menjalankan resep.

2. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.

Resep baru ini berjalan saat Anda memperbarui buku masak pada instance dan kemudian menjalankan resep baru dari dalam buku masak yang diperbarui. Langkah selanjutnya menjelaskan cara melakukannya.

Setelah Anda menyelesaikan langkah berikutnya, Anda akan dapat masuk ke instance dan kemudian mengetik emacs dari command prompt untuk meluncurkan GNU Emacs. (Untuk informasi selengkapnya, lihat Connect to Your Linux Instance.) Untuk keluar dari GNU Emacs, tekan Ctrl+X, lalu Ctrl+C.

> **Important**
>
> Untuk masuk ke instance, Anda harus terlebih dahulu memberikan AWS OpsWorks Stacks informasi tentang kunci SSH publik Anda (yang dapat Anda buat dengan alat seperti ssh-keygen atau PuTTYgen), dan kemudian Anda harus mengatur izin pada `MyCookbooksDemoStack` tumpukan untuk memungkinkan pengguna Anda masuk ke instance. Lihat petunjuknya di Mendaftarkan Kunci SSH Publik Pengguna dan Login dengan SSH.

Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak pada instance dan kemudian jalankan resep dari dalam buku masak yang diperbarui pada instance. Sepanjang sisa panduan ini, Anda mengulangi langkah ini setiap kali Anda memperbarui buku masak dengan menambahkan resep baru.

Untuk memperbarui buku masak pada instance

1. Pada panel navigasi, pilih Stack. Halaman MyCookbooksDemoStack ditampilkan.
2. Pilih Run Command. Halaman Run Command ditampilkan.

3. Untuk Perintah, pilih Perbarui Buku Masak Kustom.
4. Tinggalkan pengaturan default berikut:
   - Komentar (kosong)
   - Advanced, Custom Chef JSON (kosong)
   - Advanced, Instance (Pilih semua dicentang, MyCookbooksDemoLayerdicentang, buku masak-demo1 dicentang)
5. Pilih Perbarui Buku Masak Kustom. Halaman Running command update_custom_cookbooks ditampilkan. Jangan lanjutkan sampai Status berubah menjadi berhasil. Proses ini mungkin memakan waktu beberapa menit, jadi bersabarlah.

Untuk menjalankan resep

1. Pada panel navigasi, pilih Stack. Halaman MyCookbooksDemoStack ditampilkan.
2. Pilih Run Command. Halaman Run Command ditampilkan.
3. Untuk Command, pilih Execute Recipes.
4. Agar Resep dapat dijalankan, ketikkan nama resep yang akan dijalankan. Pertama kali Anda melakukan ini, resepnya diberi nama**`opsworks_cookbook_demo::install_package`**.

   > **Note**
   >
   > Saat Anda mengulangi prosedur ini nanti, ketikkan nama buku masak (**`opsworks_cookbook_demo`**), diikuti oleh dua titik dua (**`::`**), diikuti dengan nama resep (nama file resep, tanpa ekstensi `.rb` file).

5. Tinggalkan pengaturan default berikut:
   - Komentar (kosong)
   - Advanced, Custom Chef JSON (kosong)
   - Contoh Pilih semua dicentang, MyCookbooksDemoLayerdicentang, buku masak-demo1 dicentang)
6. Pilih Jalankan Resep. Halaman perintah Running execute_recipes ditampilkan. Jangan lanjutkan sampai Status berubah menjadi berhasil. Proses ini mungkin memakan waktu beberapa menit, jadi bersabarlah.

> **Note**
>
> Anda tidak perlu menjalankan resep secara manual. Anda dapat menetapkan resep ke peristiwa siklus hidup lapisan, seperti peristiwa Penyiapan dan Konfigurasi, dan AWS OpsWorks Tumpukan akan menjalankan resep tersebut secara otomatis saat peristiwa terjadi. Untuk informasi selengkapnya, lihat AWS OpsWorks Acara Siklus Hidup Stacks.

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk menambahkan pengguna ke instance.

Langkah 6: Perbarui Buku Masak untuk Tambahkan Pengguna

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang menambahkan pengguna lokal ke instance dan menetapkan direktori home dan shell pengguna. Ini mirip dengan menjalankan Linux adduser atau useradd perintah atau net user perintah Windows. Anda menambahkan pengguna lokal ke sebuah instance, misalnya, ketika Anda ingin mengontrol akses ke file dan direktori instans.

Anda juga dapat mengelola pengguna tanpa menggunakan buku masak. Untuk informasi selengkapnya, lihat Mengelola Pengguna.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `add_user.rb` dengan kode berikut (untuk informasi lebih lanjut, buka pengguna):

   ```
   user "Add a user" do
     home "/home/jdoe"
     shell "/bin/bash"
   ```

```
  username "jdoe"
end
```

2. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur "Menjalankan resep", agar Resep dijalankan, ketik**`opsworks_cookbook_demo::add_user`**.

Untuk menguji resepnya

1. Masuk ke instans, jika Anda belum melakukannya.
2. Dari prompt, jalankan perintah berikut untuk mengonfirmasi bahwa pengguna baru telah ditambahkan:

   ```
   grep jdoe /etc/passwd
   ```

   Informasi yang mirip dengan berikut ini ditampilkan tentang pengguna, termasuk rincian seperti nama pengguna, nomor ID, nomor ID grup, direktori home, dan shell:

   ```
   jdoe:x:501:502::/home/jdoe:/bin/bash
   ```

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk membuat direktori pada instance.

Langkah 7: Perbarui Buku Masak untuk Membuat Direktori

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang menambahkan direktori ke instance. Ini mirip dengan menjalankan mkdir perintah Linux atau Windows md atau mkdir perintah.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `create_directory.rb` dengan kode berikut. Untuk informasi lebih lanjut, buka direktori:

   ```
   directory "Create a directory" do
     group "root"
     mode "0755"
     owner "ec2-user"
     path "/tmp/create-directory-demo"
   end
   ```

2. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::create_directory`**.

Untuk menguji resepnya

1. Masuk ke instans, jika Anda belum melakukannya.
2. Dari prompt, jalankan perintah berikut untuk mengonfirmasi bahwa direktori baru telah ditambahkan:

   ```
   ls -la /tmp/create-directory-demo
   ```

   Informasi tentang direktori yang baru ditambahkan ditampilkan, termasuk informasi seperti izin, nama pemilik, dan nama grup:

   ```
   drwxr-xr-x 2 ec2-user root 4096 Nov 18 00:35 .
   drwxrwxrwt 6 root     root 4096 Nov 24 18:17 ..
   ```

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk membuat file pada instance.

Langkah 8: Perbarui Buku Masak untuk Membuat dan Menyalin File

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang menambahkan dua file ke instance. Sumber daya pertama dalam resep membuat file sepenuhnya dengan kode resep. Ini mirip dengan menjalankan Linuxcat,echo, atau touch perintah atau Windows echo atau fsutil perintah. Teknik ini berguna untuk beberapa file kecil, atau sederhana. Sumber daya kedua dalam resep menyalin file di buku masak ke direktori lain pada instance. Ini mirip dengan menjalankan cp perintah Linux atau copy perintah Windows.Teknik ini berguna untuk banyak file besar, atau kompleks.

Sebelum Anda memulai langkah ini, selesaikan Langkah 7: Perbarui Buku Masak untuk Membuat Direktori untuk memastikan bahwa direktori induk file sudah ada.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `opsworks_cookbook_demo` direktori, buat subdirektori bernama. `files`
2. Di `files` subdirektori, buat file bernama `hello.txt` dengan teks berikut: **`Hello, World!`**
3. Di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `create_files.rb` dengan kode berikut. Untuk informasi lebih lanjut, buka file dan 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
```

Sumber `file` daya membuat file di jalur yang ditentukan. `cookbook_file`Sumber daya menyalin file dari `files` direktori yang baru saja Anda buat di buku masak (Chef mengharapkan untuk menemukan subdirektori bernama standar bernama `files` yang dapat menyalin file dari) ke direktori lain pada instance.

4. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
5. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
6. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::create_files`**.

Untuk menguji resepnya

1. Masuk ke instans, jika Anda belum melakukannya.
2. Dari prompt, jalankan perintah berikut, satu per satu, untuk mengonfirmasi bahwa file baru ditambahkan:

```
sudo cat /tmp/create-directory-demo/index.html

sudo cat /tmp/create-directory-demo/hello.txt
```

Isi file ditampilkan:

```
<html>This is a placeholder for the home page.</html>

Hello, World!
```

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk menjalankan perintah pada instance.

Langkah 9: Perbarui Buku Masak untuk Menjalankan Perintah

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang menjalankan perintah yang membuat kunci SSH pada instance.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `run_command.rb` dengan kode berikut. Untuk informasi lebih lanjut, buka eksekusi.

   ```
   execute "Create an SSH key" do
     command "ssh-keygen -f /tmp/my-key -N fLyC3jbY"
   end
   ```

2. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::run_command`**.

Untuk menguji resepnya

1. Masuk ke instans, jika Anda belum melakukannya.
2. Dari prompt, jalankan perintah berikut, satu per satu, untuk mengonfirmasi bahwa kunci SSH telah dibuat:

```
sudo cat /tmp/my-key

sudo cat /tmp/my-key.pub
```

   Konten kunci pribadi dan publik SSH ditampilkan:

```
-----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
```

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk menjalankan skrip pada instance.

Langkah 10: Perbarui Cookbook untuk Menjalankan Script

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang menjalankan skrip pada instance. Resep ini membuat direktori dan kemudian membuat file di direktori itu. Menulis resep untuk menjalankan skrip yang berisi banyak perintah lebih mudah daripada menjalankan perintah itu satu per satu.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `run_script.rb` dengan kode berikut. Untuk informasi lebih lanjut, buka skrip.

   ```
   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. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur "Menjalankan resep", agar Resep dijalankan, ketik**`opsworks_cookbook_demo::run_script`**.

Untuk menguji resepnya

1. Masuk ke instans, jika Anda belum melakukannya.
2. Dari prompt, jalankan perintah berikut untuk mengonfirmasi bahwa file baru telah ditambahkan:

   ```
   sudo cat /tmp/run-script-demo/helloworld.txt
   ```

   Isi file ditampilkan:

   ```
   Hello, World!
   ```

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk mengelola layanan pada instance.

Langkah 11: Perbarui Buku Masak untuk Mengelola Layanan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang mengelola layanan pada instance. Ini mirip dengan menjalankan service perintah Linux atau Windowsnet stop,net start, dan perintah serupa. Resep ini menghentikan crond layanan pada instance.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `manage_service.rb` dengan kode berikut. Untuk informasi lebih lanjut, buka layanan.

   ```
   service "Manage a service" do
     action :stop
     service_name "crond"
   end
   ```

2. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::manage_service`**.

Untuk menguji resepnya

1. Masuk ke instans, jika Anda belum melakukannya.

2. Dari prompt, jalankan perintah berikut untuk mengonfirmasi bahwa crond layanan dihentikan:

```
service crond status
```

Berikut ini ditampilkan secara default.

```
crond is stopped
```

3. Untuk memulai ulang crond layanan, jalankan perintah berikut:

```
sudo service crond start
```

Berikut ini ditampilkan secara default.

```
Starting crond:  [  OK  ]
```

4. Untuk mengonfirmasi bahwa crond layanan telah dimulai, jalankan perintah berikut lagi:

```
service crond status
```

Informasi yang serupa dengan hal berikut ini ditampilkan secara default.

```
crond (pid  3917) is running...
```

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk referensi informasi yang disimpan sebagai JSON khusus pada instance.

Langkah 12: Perbarui Buku Masak untuk Menggunakan JSON Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang mereferensikan JSON khusus yang disimpan pada instance.

Anda dapat menentukan informasi dalam format JSON kustom setiap kali Anda membuat, memperbarui, atau mengkloning tumpukan atau ketika Anda menjalankan perintah deployment atau stack. Ini berguna, misalnya, untuk membuat sebagian kecil data yang tidak berubah tersedia untuk resep Anda pada instance alih-alih mendapatkan data ini dari database. Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom.

Untuk panduan ini, Anda akan menggunakan JSON khusus untuk memberikan beberapa informasi fiktif tentang faktur pelanggan. JSON kustom dijelaskan nanti dalam langkah ini.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `custom_json.rb` yang berisi kode resep berikut:

   ```
   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']}' **********")
   ```

   Resep ini menampilkan pesan di log tentang nilai di JSON kustom.
2. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::custom_json`**. Untuk Advanced, Custom Chef JSON, ketik JSON kustom berikut:

   ```
   {
     "customer-id": "0123",
   ```

```
  "invoice-number": "9876",
  "line-items": {
    "line-1": "tractor",
    "line-2": "passenger car",
    "line-3": "trailer"
  }
}
```

Untuk menguji resepnya

1. Dengan perintah Running execute_recipes halaman ditampilkan dari prosedur sebelumnya, untuk cookbooks-demo1, untuk Log, pilih show. Halaman log execute_recipes ditampilkan.
2. Gulir ke bawah melalui log untuk menemukan entri yang terlihat serupa dengan berikut ini:

```
[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'
 **********
```

   Entri ini menampilkan informasi dari JSON kustom yang diketik di kotak Advanced, Custom Chef JSON.

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk mendapatkan informasi dari tas data, yang merupakan kumpulan pengaturan tumpukan yang disimpan AWS OpsWorks Stacks pada setiap instance.

Langkah 13: Perbarui Buku Masak untuk Menggunakan Tas Data

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang mereferensikan pengaturan tumpukan yang disimpan AWS OpsWorks Stacks pada instance dalam satu set kantong data. Resep ini menampilkan pesan di log tentang pengaturan tumpukan tertentu yang disimpan pada instance. Untuk informasi selengkapnya, lihat AWS OpsWorks Referensi Tas Data Stacks.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `data_bags.rb` yang berisi kode berikut:

   ```
   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']}' **********")
   ```

   Resep ini menampilkan pesan di log tentang pengaturan tumpukan tertentu yang disimpan pada instance.

2. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.

3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.

4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur "Menjalankan resep", agar Resep dijalankan, ketik**`opsworks_cookbook_demo::data_bags`**.

Untuk menguji resepnya

1. Dengan perintah Running execute_recipes halaman ditampilkan dari prosedur sebelumnya, untuk cookbooks-demo1, untuk Log, pilih show. Halaman log execute_recipes ditampilkan.
2. Gulir ke bawah melalui log dan temukan entri yang terlihat serupa dengan berikut ini:

```
[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'
 **********
```

   Resep ini menampilkan pesan tentang pengaturan tumpukan tertentu yang disimpan pada instance.

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk menjalankan kode resep beberapa kali.

Langkah 14: Perbarui Cookbook untuk Menggunakan Iterasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Perbarui buku masak Anda dengan menambahkan resep yang menggunakan iterasi, teknik yang mengulangi kode resep beberapa kali. Resep ini menampilkan pesan di log untuk item tas data yang berisi beberapa konten.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `iteration_demo.rb` yang berisi kode berikut:

```
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**
>
> Menulis kode resep sebelumnya lebih pendek, lebih fleksibel, dan kurang rawan kesalahan daripada menulis kode resep berikut yang tidak menggunakan iterasi:
>
> ```
> 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. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan

resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::iteration_demo`**.

Untuk menguji resepnya

1. Dengan perintah Running execute_recipes halaman ditampilkan dari prosedur sebelumnya, untuk cookbooks-demo1, untuk Log, pilih show. Halaman log execute_recipes ditampilkan.
2. Gulir ke bawah melalui log dan temukan entri yang terlihat serupa dengan berikut ini:

```
[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]' **********
```

Resep ini menampilkan pesan di log untuk item tas data yang berisi beberapa konten. Item tas data ada di tas `aws_opsworks_stack` data. Item tas data memiliki konten bernama`custom_cookbooks_source`. Di dalam konten ini ada enam konten bernama`type`, `url`,`username`,`password`,`ssh_key`,, dan`revision`; nilainya juga ditampilkan.

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk menjalankan kode resep hanya jika kondisi tertentu terpenuhi.

Langkah 15: Perbarui Buku Masak untuk Menggunakan Logika Bersyarat

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang perbarui buku masak Anda dengan menambahkan resep yang menggunakan logika bersyarat, teknik yang menjalankan kode hanya jika kondisi tertentu terpenuhi. Untuk informasi lebih lanjut, buka Pernyataan if dan Pernyataan kasus.

Resep ini melakukan dua hal berdasarkan konten kantong data: menampilkan pesan di log yang mengidentifikasi sistem operasi tempat instance berjalan dan, hanya jika sistem operasinya Linux, menginstal paket dengan menggunakan manajer paket yang benar untuk distribusi Linux yang diberikan. Paket ini diberi nama pohon; ini adalah aplikasi sederhana untuk memvisualisasikan daftar direktori.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di workstation lokal Anda, di `recipes` subdirektori di`opsworks_cookbook_demo` `directory`, buat file bernama `conditional_logic.rb` yang berisi kode berikut:

```
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. Di terminal atau command prompt, gunakan tar perintah buat versi baru `opsworks_cookbook_demo.tar.gz` file, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui.
3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::conditional_logic`**.

Untuk menguji resepnya

1. Dengan perintah Running execute_recipes halaman ditampilkan dari prosedur sebelumnya, untuk cookbooks-demo1, untuk Log, pilih show. Halaman log execute_recipes ditampilkan.
2. Gulir ke bawah melalui log dan temukan entri yang terlihat serupa dengan yang berikut ini:

   ```
   [2015-11-16T19:59:05+00:00] INFO: ********** Operating system is Amazon Linux.
    **********
   ```

   Karena sistem operasi instans adalah Amazon Linux 2016.09, hanya entri sebelumnya (dari lima entri yang mungkin dalam kode resep) yang akan ditampilkan di log.
3. Jika sistem operasinya Linux, resepnya menginstal paket pohon. Untuk melihat visualisasi isi direktori, ketik **`tree`** prompt perintah dari direktori yang diinginkan atau dengan jalur direktori yang diinginkan (misalnya,`tree /var/chef/runs`).

Pada langkah berikutnya, Anda akan memperbarui buku masak untuk menggunakan fungsionalitas dari buku masak eksternal yang disediakan oleh komunitas Chef.

Langkah 16: Perbarui Buku Masak untuk Menggunakan Buku Masak Komunitas

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Terakhir, perbarui buku masak untuk menggunakan fungsionalitas yang disediakan dalam buku masak eksternal yang disediakan oleh komunitas Chef. Buku masak eksternal yang akan Anda gunakan untuk panduan ini tersedia melalui Chef Supermarket, lokasi populer untuk mengakses buku masak Chef eksternal. Buku masak eksternal ini menyediakan sumber daya khusus yang memungkinkan Anda mengunduh dan menginstal aplikasi, mirip dengan apa yang Anda lakukan. Langkah 4: Memperbarui Buku Masak untuk Instal Package Namun, sumber daya ini dapat menginstal aplikasi web dan jenis aplikasi lain selain paket.

Ketika buku masak tergantung pada buku masak lain, Anda harus menentukan dependensi pada buku masak lainnya. Untuk mendeklarasikan dan mengelola dependensi buku masak, kami menyarankan Anda untuk menggunakan alat yang disebut Berkshelf. Untuk informasi selengkapnya tentang cara menginstal Berkshelf di workstation lokal Anda, lihat Tentang Berkshelf di situs web Chef.

Setelah Anda menginstal Berkshelf, ikuti prosedur ini untuk mendeklarasikan ketergantungan buku masak dan kemudian buat resep yang memanggil sumber daya di buku masak eksternal:

Untuk mendeklarasikan ketergantungan buku masak

1. Di workstation lokal Anda, di `opsworks_cookbook_demo` direktori, tambahkan baris berikut di akhir `metadata.rb` file:

   ```
   depends "application", "5.0.0"
   ```

   Ini menyatakan dependensi pada buku masak bernama`application`, versi 5.0.0.
2. Dari root `opsworks_cookbook_demo` direktori, jalankan perintah berikut. Periode di akhir perintah disengaja.

```
berks init .
```

Berkshelf membuat sejumlah folder dan file yang dapat Anda gunakan nanti untuk skenario yang lebih canggih. Satu-satunya file yang kita butuhkan untuk panduan ini adalah file bernama `Berksfile`

3. Tambahkan baris berikut di akhir `Berksfile` file:

```
cookbook "application", "5.0.0"
```

Ini memberi tahu Berkshelf bahwa Anda ingin menggunakan buku masak aplikasi versi 5.0.0, yang diunduh Berkshelf dari Chef Supermarket.

4. Pada terminal atau prompt, jalankan perintah berikut dari root `opsworks_cookbook_demo` direktori:

```
berks install
```

Berkshelf membuat daftar dependensi untuk buku masak Anda dan buku masak aplikasi. Berkshelf menggunakan daftar dependensi ini dalam prosedur berikutnya.

Untuk memperbarui buku masak pada instance dan menjalankan resep baru

1. Di `recipes` subdirektori di `opsworks_cookbook_demo` direktori, buat file bernama `dependencies_demo.rb` yang berisi kode berikut:

```
application "Install NetHack" do
  package "nethack.x86_64"
end
```

Resep ini tergantung pada sumber daya aplikasi dari buku masak aplikasi untuk menginstal game petualangan berbasis teks populer NetHack pada instance. (Anda dapat, tentu saja, subtitute nama paket lain yang Anda inginkan, asalkan paket sudah tersedia untuk manajer paket pada instance.)

2. Dari root `opsworks_cookbook_demo` direktori, jalankan perintah berikut:

```
berks package
```

Berkshelf menggunakan daftar dependensi dari prosedur sebelumnya untuk membuat file bernama`cookbooks-`*`timestamp`*`.tar.gz`, yang berisi `opsworks_cookbook_demo` direktori dan isinya yang diperbarui, termasuk buku masak tergantung buku masak. Ubah nama file `opsworks_cookbook_demo.tar.gz` ini.

3. Unggah `opsworks_cookbook_demo.tar.gz` file yang diperbarui dan diganti namanya ke bucket S3 Anda.
4. Ikuti prosedur Langkah 5: Perbarui Buku Masak pada Instance dan Jalankan Resepnya untuk memperbarui buku masak pada instance dan menjalankan resepnya. Dalam prosedur “Menjalankan resep”, agar Resep dijalankan, ketik**`opsworks_cookbook_demo::dependencies_demo`**.
5. Setelah Anda menjalankan resep, Anda harus dapat masuk ke instance dan kemudian mengetik **`nethack`** pada command prompt untuk mulai bermain. (Untuk informasi lebih lanjut tentang permainan, lihat NetHackdan NetHackBuku Panduan.)

Pada langkah selanjutnya, Anda dapat membersihkan AWS sumber daya yang Anda gunakan untuk panduan ini. Langkah selanjutnya bersifat opsional. Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks. Namun, menyimpan AWS sumber daya ini dapat mengakibatkan beberapa tagihan berkelanjutan ke AWS akun Anda. Jika Anda ingin menyimpan AWS sumber daya ini untuk digunakan nanti, Anda sekarang telah menyelesaikan panduan ini, dan Anda dapat melompat ke depan. Langkah Berikutnya

Langkah 17: (Opsional) Hapus

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk mencegah timbulnya biaya tambahan ke AWS akun Anda, Anda dapat menghapus AWS sumber daya yang digunakan untuk panduan ini. AWS Sumber daya ini termasuk bucket S3, tumpukan AWS OpsWorks Stacks, dan komponen tumpukan. (Untuk informasi selengkapnya, lihat OpsWorks Harga AWS.) Namun, Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat

Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks. Jika Anda ingin menyimpan AWS sumber daya ini tersedia, Anda sekarang telah menyelesaikan panduan ini, dan Anda dapat melompat ke. Langkah Berikutnya

Konten yang disimpan dalam sumber daya yang Anda buat untuk panduan ini dapat berisi informasi identifikasi pribadi. Jika Anda tidak lagi ingin informasi ini disimpan AWS, ikuti langkah-langkah dalam topik ini.

Untuk menghapus bucket S3

- Lihat Hapus Bucket Amazon S3.

Untuk menghapus instans untuk tumpukan

1. Di konsol AWS OpsWorks Stacks, di panel navigasi layanan, pilih Instans. Halaman Instans ditampilkan.
2. Untuk MyCookbooksDemoLayer, untuk buku masak-demo1, untuk Tindakan, pilih berhenti. Saat Anda melihat pesan konfirmasi, pilih Berhenti.
3. Perubahan berikut terjadi selama beberapa menit. Jangan lanjutkan sampai semua hal berikut selesai.
   - Status berubah dari online menjadi berhenti dan akhirnya berhenti.
   - perubahan online dari 1 menjadi 0.
   - mematikan perubahan dari 0 ke 1 dan akhirnya kembali ke 0.
   - berhenti akhirnya berubah dari 0 ke 1.
4. Untuk Tindakan, pilih hapus. Saat Anda melihat pesan konfirmasi, pilih Hapus. AWS OpsWorks Tumpukan menghapus instance dan menampilkan No instance.

Untuk menghapus tumpukan

1. Pada panel navigasi, pilih Stack. Halaman MyCookbooksDemoStack ditampilkan.
2. Pilih Hapus Tumpukan. Saat Anda melihat pesan konfirmasi, pilih Hapus. AWS OpsWorks Stacks menghapus tumpukan dan menampilkan halaman Dashboard.

Secara opsional, Anda dapat menghapus pengguna IAM dan EC2 key pair Amazon yang Anda gunakan untuk panduan ini, jika Anda tidak ingin menggunakannya kembali untuk akses ke layanan

dan instance lain. AWS EC2 Untuk petunjuk, lihat Menghapus pengguna IAM dan pasangan EC2 kunci Amazon dan instance Linux.

Anda sekarang telah menyelesaikan panduan ini. Untuk informasi selengkapnya, lihat Langkah Berikutnya.

Langkah Berikutnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang setelah Anda menyelesaikan panduan ini, Anda dapat mempelajari lebih lanjut tentang dukungan AWS OpsWorks Stacks untuk buku masak Chef dengan meninjau sumber daya berikut:

- Buku Masak dan Resep— Menjelaskan versi Chef dan Ruby yang didukung AWS OpsWorks Stacks saat ini. Juga menunjukkan cara menginstal dan memperbarui buku masak khusus pada instance dan cara menjalankan resep pada instance.
- Belajar Chef — Menyediakan tautan ke tutorial Chef, perpustakaan keterampilan Chef, dokumentasi Chef lengkap, dan kelas pelatihan Chef.
- Semua tentang Chef — Menyediakan dokumentasi Chef lengkap. Topik khusus yang menarik meliputi:
  - Tentang Buku Masak — Menjelaskan komponen buku masak utama seperti atribut, resep, file, metadata, dan templat.
  - Tentang Resep — Menjelaskan dasar-dasar resep seperti cara bekerja dengan kantong data, memasukkan resep lain, dan menggunakan kode Ruby dalam resep.
  - Sumber Daya — Menjelaskan cara menggunakan semua sumber daya Chef bawaan, sepertiapt_package,cookbook_file,directory,execute,file, danpackage.
  - Tentang Resep DSL — Menjelaskan cara menulis kode untuk resep Chef dengan pernyataan sepertiif,,case, data_bagdata_bag_item, dansearch.
- Tentang Template - Menjelaskan cara menggunakan templat Embedded Ruby (ERB) untuk menghasilkan file teks statis secara dinamis, seperti file konfigurasi.

- Learning Tracks — Menjelaskan cara menggunakan Chef untuk mengelola instans, mengelola aplikasi web dasar, mengembangkan dan menguji kode infrastruktur, menggunakan analitik Chef, dan banyak lagi.
- Learning Chef — Pengantar Chef. Diterbitkan oleh O'Reilly Media.
- Contoh kode Learning Chef — Memberikan contoh kode untuk menemani buku Learning Chef yang diterbitkan oleh O'Reilly Media.

AWS OpsWorks Praktik Terbaik Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Strategi, teknik, dan saran di bagian ini dapat membantu Anda mendapatkan manfaat maksimal dan hasil optimal dari AWS OpsWorks Stacks.

Topik

- Praktik Terbaik: Penyimpanan Perangkat Root untuk Instans
- Praktik Terbaik: Mengoptimalkan Jumlah Server Aplikasi
- Praktik Terbaik: Mengelola Izin
- Praktik Terbaik: Mengelola dan Menyebarkan Aplikasi dan Buku Masak
- Ketergantungan Buku Masak Kemasan Secara Lokal

Praktik Terbaik: Penyimpanan Perangkat Root untuk Instans

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Topik ini tidak berlaku untuk instance Windows, yang harus didukung Amazon Elastic Block Store.

Instans Amazon Elastic Compute Cloud (Amazon EC2) Linux memiliki opsi penyimpanan perangkat root berikut.

- Instans yang didukung toko instans — Perangkat root bersifat sementara.

  Jika Anda menghentikan instance, data pada perangkat root menghilang dan tidak dapat dipulihkan. Untuk informasi selengkapnya, lihat Amazon EC2 Instance Store.
- Instans yang didukung Amazon EBS — Perangkat root adalah volume Amazon EBS.

  Jika Anda menghentikan instans, volume Amazon EBS tetap ada. Jika Anda me-restart instance, volume secara otomatis di-remount, memulihkan status instance dan data yang disimpan. Anda juga dapat memasang volume pada instance yang berbeda. Untuk informasi selengkapnya, lihat Amazon Elastic Block Store (Amazon EBS).

Pertimbangkan hal berikut saat memutuskan opsi penyimpanan perangkat root mana yang akan digunakan.

Waktu Boot

Setelah start awal, instans Amazon EBS umumnya restart lebih cepat.

Waktu startup awal kira-kira sama untuk kedua jenis penyimpanan. Kedua jenis harus melakukan pengaturan penuh, yang mencakup tugas yang relatif memakan waktu seperti menginstal paket dari repositori jarak jauh. Namun, perhatikan perbedaan ini saat Anda memulai ulang instance:

- Instance yang didukung toko instans melakukan tugas penyiapan yang sama seperti yang mereka lakukan untuk memulai awal, termasuk penginstalan paket.

  Restart membutuhkan waktu yang hampir sama dengan awal awal.

- Instans Amazon EBS-back mengubah volume root dan menjalankan resep Setup.

  Restart biasanya secara signifikan lebih cepat daripada awal awal, karena resep Setup tidak harus melakukan tugas-tugas seperti menginstal ulang paket yang sudah diinstal pada volume root.

Biaya

Instans yang didukung Amazon EBS lebih mahal:

- Dengan instans yang didukung instance-store, Anda hanya membayar saat instance sedang berjalan.
- Dengan instans yang didukung Amazon EBS, Anda membayar volume Amazon EBS baik instans berjalan atau tidak.

  Untuk informasi selengkapnya, lihat Harga Amazon EBS.

Pencatatan log

Instans yang didukung Amazon EBS secara otomatis menyimpan log:

- Dengan instance yang didukung toko instance, log menghilang saat instance berhenti.

  Anda harus mengambil log sebelum menghentikan instance atau menggunakan layanan seperti CloudWatch Log untuk menyimpan log yang dipilih dari jarak jauh.
- Dengan instans yang didukung Amazon EBS, log disimpan pada volume Amazon EBS.

  Anda dapat melihatnya dengan memulai ulang instance, atau dengan memasang volume pada instance lain.

Dependensi

Kedua jenis penyimpanan memiliki dependensi yang berbeda:

- Instans yang didukung instance-store bergantung pada Amazon S3.

  Saat Anda memulai instance, itu harus mengunduh AMI dari Amazon S3.
- Instans yang didukung Amazon EBS bergantung pada Amazon EBS.

  Saat Anda memulai instance, instans harus memasang volume root Amazon EBS.

Rekomendasi: Jika Anda tidak yakin jenis penyimpanan mana yang paling sesuai dengan kebutuhan Anda, sebaiknya mulai dengan instans Amazon EBS. Meskipun Anda akan dikenakan biaya sederhana untuk volume Amazon EBS, ada sedikit risiko kehilangan data yang tidak diinginkan.

Praktik Terbaik: Mengoptimalkan Jumlah Server Aplikasi

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan produksi biasanya mencakup beberapa server aplikasi yang didistribusikan di beberapa Availability Zone. Namun jumlah permintaan yang masuk dapat sangat bervariasi tergantung pada waktu hari atau hari dalam seminggu. Anda bisa menjalankan server yang cukup untuk menangani beban maksimum yang diantisipasi, tetapi kemudian sebagian besar waktu Anda akhirnya akan membayar lebih banyak kapasitas server daripada yang Anda butuhkan. Untuk menjalankan situs Anda secara efisien, praktik yang disarankan adalah mencocokkan jumlah server dengan volume permintaan saat ini.

AWS OpsWorks Stacks menyediakan tiga cara untuk mengelola jumlah instance server.

- Instance 24/7 dimulai secara manual dan dijalankan hingga dihentikan secara manual.
- Instans berbasis waktu secara otomatis dimulai dan dihentikan oleh AWS OpsWorks Stacks pada jadwal yang ditentukan pengguna.
- Instans berbasis beban secara otomatis dimulai dan dihentikan oleh AWS OpsWorks Stacks ketika melewati ambang batas untuk metrik beban yang ditentukan pengguna seperti CPU atau pemanfaatan memori.

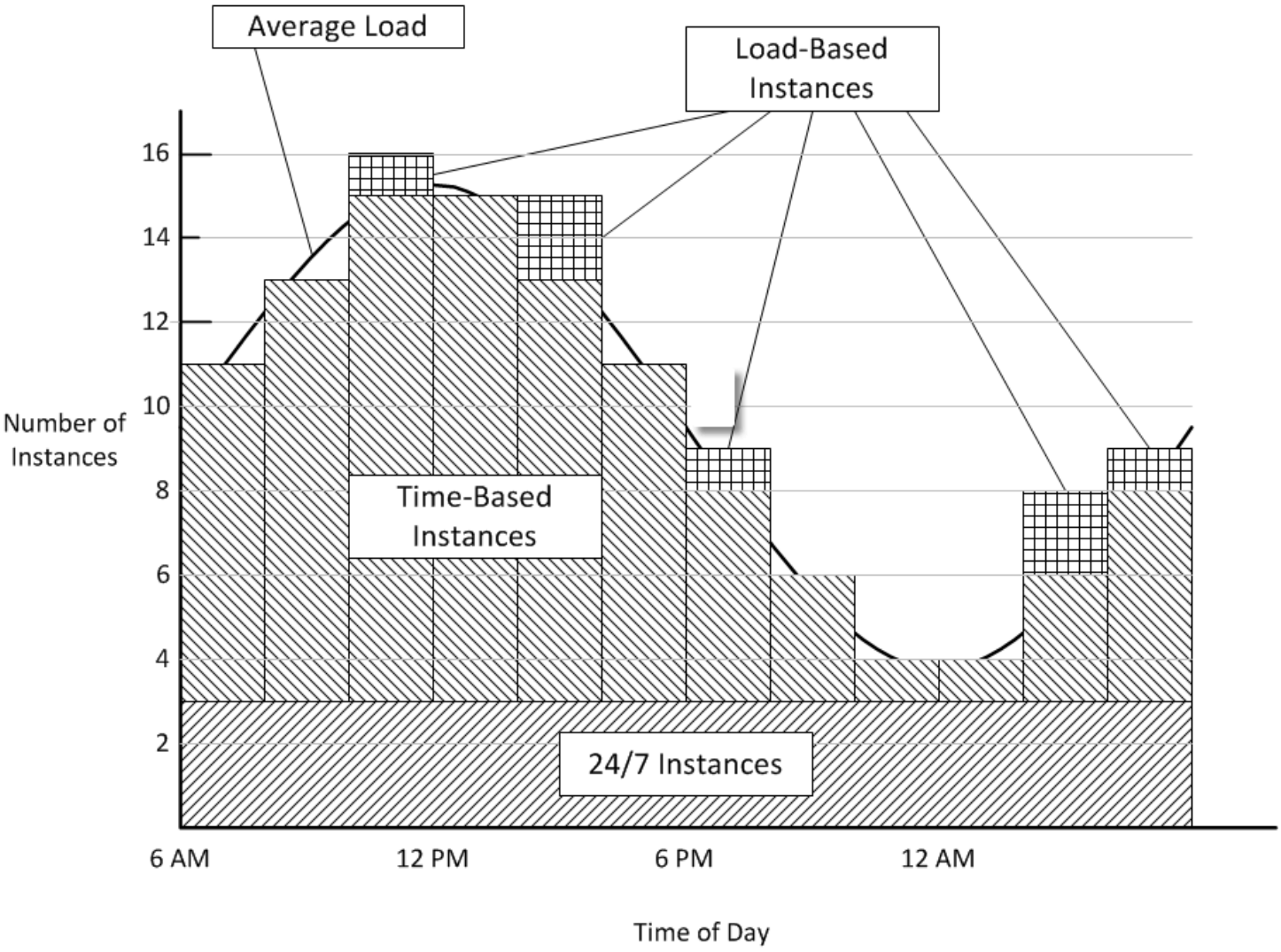
Note

Setelah Anda membuat dan mengonfigurasi waktu tumpukan dan instance berbasis beban, AWS OpsWorks Stacks secara otomatis memulai dan menghentikannya berdasarkan konfigurasi yang ditentukan. Anda tidak perlu menyentuhnya lagi kecuali Anda memutuskan untuk mengubah konfigurasi atau jumlah instance.

Rekomendasi: Jika Anda mengelola tumpukan dengan lebih dari beberapa instance server aplikasi, sebaiknya gunakan campuran ketiga jenis instans. Berikut ini adalah contoh bagaimana mengelola kapasitas server stack untuk menangani volume permintaan harian variabel dengan karakteristik sebagai berikut.

- Volume permintaan rata-rata bervariasi secara sinusoidal sepanjang hari.
- Volume permintaan rata-rata minimum membutuhkan lima instance server aplikasi.
- Volume permintaan rata-rata maksimum membutuhkan enam belas instance server aplikasi.
- Lonjakan volume permintaan biasanya dapat ditangani oleh satu atau dua instance server aplikasi.

Ini adalah model yang nyaman untuk tujuan diskusi, tetapi Anda dapat dengan mudah menyesuaikannya dengan variasi volume permintaan dan juga memperluasnya untuk menangani variasi mingguan. Diagram berikut menunjukkan cara menggunakan tiga jenis instance untuk mengelola volume permintaan ini.



Contoh ini memiliki karakteristik sebagai berikut:

- Tumpukan memiliki tiga instance 24/7, yang selalu aktif dan menangani beban dasar.
- Tumpukan memiliki 12 instance berbasis waktu, yang dikonfigurasi untuk menangani variasi harian rata-rata.

  Satu berjalan dari jam 10 malam hingga 2 pagi, dua lagi lari dari jam 8 malam hingga 10 malam dan 2 pagi hingga 4 pagi, dan seterusnya. Untuk mempermudah, diagram memodifikasi jumlah instance berbasis waktu setiap dua jam, tetapi Anda dapat memodifikasi angka setiap jam jika Anda menginginkan kontrol yang lebih halus.
- Tumpukan memiliki instance berbasis beban yang cukup untuk menangani lonjakan lalu lintas yang melebihi apa yang dapat ditangani oleh instance 24/7 dan berbasis waktu.

AWS OpsWorks Tumpukan memulai instance berbasis beban hanya jika beban di semua server yang sedang berjalan melebihi metrik yang ditentukan. Biaya untuk instans yang tidak berjalan minimal (instans yang didukung Amazon EBS) atau tidak sama sekali (instans yang didukung toko instans), jadi praktik yang disarankan adalah membuat cukup banyak instans untuk menangani volume permintaan maksimum yang diantisipasi dengan nyaman. Untuk contoh ini, tumpukan harus memiliki setidaknya tiga instance berbasis beban.

> **Note**
>
> Pastikan Anda memiliki ketiga jenis instans yang didistribusikan di beberapa Availability Zone untuk mengurangi dampak gangguan layanan apa pun.

Praktik Terbaik: Mengelola Izin

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda harus memiliki beberapa bentuk kredensi AWS untuk mengakses sumber daya akun Anda. Berikut ini adalah beberapa pedoman umum untuk menyediakan akses ke karyawan Anda.

- Pertama dan terpenting, kami menyarankan Anda untuk tidak menggunakan kredensi root akun Anda untuk mengakses sumber daya AWS.

  Sebagai gantinya, buat Identitas IAM untuk karyawan Anda dan tambahkan izin yang menyediakan akses yang sesuai. Setiap karyawan kemudian dapat menggunakan kredensialnya untuk mengakses sumber daya.
- Karyawan harus memiliki izin untuk mengakses hanya sumber daya yang mereka butuhkan untuk melakukan pekerjaan mereka.

Misalnya, pengembang aplikasi hanya perlu mengakses tumpukan yang menjalankan aplikasi mereka.

- Karyawan harus memiliki izin untuk hanya menggunakan tindakan yang mereka butuhkan untuk melakukan pekerjaan mereka.

  Pengembang aplikasi mungkin memerlukan izin penuh untuk tumpukan pengembangan dan izin untuk menerapkan aplikasi mereka ke tumpukan produksi yang sesuai. Mereka mungkin tidak memerlukan izin untuk memulai atau menghentikan instance pada tumpukan produksi, membuat atau menghapus lapisan, dan sebagainya.

Untuk informasi umum selengkapnya tentang mengelola izin, lihat AWS Security Credentials.

Anda dapat menggunakan AWS OpsWorks Stacks atau IAM untuk mengelola izin pengguna. Perhatikan bahwa kedua opsi tidak saling eksklusif; kadang-kadang diinginkan untuk menggunakan keduanya.

AWS OpsWorks Manajemen Izin Tumpukan

Setiap tumpukan memiliki halaman Izin yang dapat Anda gunakan untuk memberikan izin kepada pengguna untuk mengakses tumpukan dan menentukan tindakan apa yang dapat mereka lakukan. Anda menentukan izin pengguna dengan menyetel salah satu tingkat izin berikut. Setiap level mewakili kebijakan IAM yang memberikan izin untuk serangkaian tindakan standar.

- Tolak menolak izin untuk berinteraksi dengan tumpukan dengan cara apa pun.
- Tampilkan izin hibah melihat konfigurasi tumpukan tetapi tidak mengubah status tumpukan dengan cara apa pun.
- Deploy menyertakan izin Tampilkan dan juga memberikan izin pengguna untuk menerapkan aplikasi.
- Kelola mencakup izin Deploy dan juga memungkinkan pengguna untuk melakukan berbagai tindakan manajemen tumpukan, seperti membuat atau menghapus instance dan lapisan.

> **Note**
>
> Tingkat Kelola izin tidak memberikan izin untuk sejumlah kecil tindakan AWS OpsWorks Tumpukan tingkat tinggi, termasuk membuat atau mengkloning tumpukan. Anda harus menggunakan kebijakan IAM untuk memberikan izin tersebut.

Selain menyetel tingkat izin, Anda juga dapat menggunakan halaman Izin tumpukan untuk menentukan apakah pengguna memiliki SSH/RDP and sudo/admin hak istimewa pada instance tumpukan. Untuk informasi selengkapnya tentang manajemen izin AWS OpsWorks Stacks, lihat. Memberikan Izin Per-Stack Untuk informasi selengkapnya tentang mengelola akses SSH, lihatMengelola Akses SSH.

Manajemen Izin IAM

Dengan manajemen izin IAM, Anda menggunakan konsol IAM, API, atau CLI untuk melampirkan kebijakan berformat JSON ke pengguna yang secara eksplisit menentukan izinnya. Untuk informasi selengkapnya tentang manajemen izin IAM, lihat Apa itu IAM? .

Rekomendasi: Mulailah dengan manajemen Izin AWS OpsWorks Tumpukan. Jika Anda perlu menyempurnakan izin pengguna, atau memberikan izin pengguna yang tidak disertakan dalam tingkat Kelola izin, Anda dapat menggabungkan kedua pendekatan tersebut. AWS OpsWorks Stacks kemudian mengevaluasi kedua kebijakan untuk menentukan izin pengguna.

Important

Jika pengguna memiliki beberapa kebijakan dengan izin yang bertentangan, penolakan selalu menang. Misalnya, Anda melampirkan kebijakan IAM ke pengguna yang mengizinkan akses ke tumpukan tertentu tetapi juga menggunakan halaman Izin tumpukan untuk menetapkan tingkat izin Tolak kepada pengguna. Tingkat izin Tolak diutamakan, dan pengguna tidak akan dapat mengakses tumpukan. Untuk informasi selengkapnya, lihat logika evaluasi kebijakan IAM.

Misalnya, Anda ingin pengguna dapat melakukan sebagian besar operasi pada tumpukan, kecuali untuk menambahkan atau menghapus lapisan.

- Tentukan tingkat izin Kelola, yang memungkinkan pengguna melakukan sebagian besar tindakan manajemen tumpukan, termasuk membuat dan menghapus lapisan.
- Lampirkan kebijakan yang dikelola pelanggan berikut ke pengguna, yang menolak izin untuk menggunakan CreateLayerdan DeleteLayertindakan pada tumpukan tersebut. Anda mengidentifikasi tumpukan dengan Nama Sumber Daya Amazon (ARN), yang dapat ditemukan di halaman Pengaturan tumpukan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:*:*:stack/2f18b4cb-4de5-4429-a149-ff7da9f0d8ee/"
    }
  ]
}
```

Untuk informasi selengkapnya, termasuk kebijakan contoh, lihat Mengelola Izin AWS OpsWorks Tumpukan dengan Melampirkan Kebijakan IAM.

> **Note**
>
> Cara lain untuk menggunakan kebijakan IAM adalah dengan menetapkan kondisi yang membatasi akses tumpukan ke karyawan dengan alamat IP atau rentang alamat tertentu. Misalnya, untuk memastikan bahwa karyawan mengakses tumpukan hanya dari dalam firewall perusahaan Anda, tetapkan kondisi yang membatasi akses ke rentang alamat IP perusahaan Anda. Untuk informasi selengkapnya, lihat Ketentuan.

Praktik Terbaik: Mengelola dan Menyebarkan Aplikasi dan Buku Masak

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks menyebarkan aplikasi dan buku masak ke setiap instance baru dari repositori jarak jauh. Selama masa hidup instans, Anda sering harus memperbarui aplikasi atau buku masak

pada instance online stack untuk menambahkan fitur, memperbaiki bug, dan sebagainya. Ada berbagai cara untuk mengelola aplikasi dan buku masak tumpukan, tetapi pendekatan yang Anda gunakan harus memenuhi persyaratan umum berikut:

- Semua instance tumpukan produksi harus memiliki aplikasi dan kode buku masak khusus yang sama, dengan pengecualian terbatas untuk tujuan seperti pengujian A/B.
- Menyebarkan pembaruan seharusnya tidak mengganggu operasi situs, bahkan jika terjadi kesalahan.

Bagian ini menjelaskan praktik yang direkomendasikan untuk mengelola dan menerapkan aplikasi dan buku masak khusus.

Topik

- Menjaga Konsistensi
- Menerapkan Kode ke Instans Online

Menjaga Konsistensi

Secara umum, Anda perlu mempertahankan kontrol ketat atas aplikasi atau kode buku masak yang berjalan di tumpukan produksi Anda. Biasanya, semua instance harus menjalankan versi kode yang saat ini disetujui. Pengecualian terjadi saat memperbarui aplikasi atau buku masak Anda, seperti yang dijelaskan nanti, dan saat mengakomodasi kasus khusus, seperti melakukan pengujian A/B.

Kode aplikasi dan buku masak disebarkan dari repositori sumber tertentu ke instance tumpukan Anda dengan dua cara:

- Saat Anda memulai sebuah instance, AWS OpsWorks Stacks secara otomatis menerapkan aplikasi dan kode buku masak saat ini ke instance.
- Untuk instance online, Anda harus secara manual menerapkan aplikasi atau kode buku masak saat ini dengan menjalankan perintah Deploy (untuk aplikasi) atau perintah Perbarui Buku Masak Kustom (untuk buku masak).

Karena ada dua mekanisme penerapan, penting bagi Anda untuk mengelola kode sumber Anda dengan hati-hati untuk menghindari menjalankan kode yang berbeda secara tidak sengaja pada instance yang berbeda. Misalnya, jika Anda menerapkan aplikasi atau buku masak dari cabang master Git, AWS OpsWorks Stacks akan menyebarkan apa yang ada di cabang tersebut pada saat

itu. Jika Anda memperbarui kode di cabang master dan kemudian memulai instance baru, instance itu akan memiliki versi kode yang lebih baru daripada instance yang lebih lama. Versi yang lebih baru bahkan mungkin tidak disetujui untuk produksi.

Rekomendasi: Arsip Amazon S3

Untuk memastikan bahwa semua instans Anda memiliki versi kode yang disetujui, sebaiknya gunakan aplikasi dan buku masak Anda dari arsip Amazon Simple Storage Service (Amazon S3). Ini menjamin bahwa kode tersebut adalah artefak statis—.zip atau file arsip lainnya—yang harus diperbarui secara eksplisit. Selain itu, Amazon S3 sangat andal, sehingga Anda jarang, jika pernah, tidak dapat mengakses arsip. Untuk memastikan konsistensi lebih lanjut, versi secara eksplisit setiap file arsip dengan menggunakan konvensi penamaan atau dengan menggunakan versi Amazon S3, yang menyediakan jejak audit dan cara mudah untuk kembali ke versi sebelumnya.

Misalnya, Anda dapat membuat pipeline penerapan menggunakan alat seperti Jenkins. Setelah kode yang ingin Anda terapkan telah dilakukan dan diuji, buat file arsip dan unggah ke Amazon S3. Semua penerapan aplikasi atau pembaruan buku masak akan menginstal kode dalam file arsip itu dan setiap instance akan memiliki kode yang sama.

Rekomendasi: Git atau Subversion Repositori

Jika Anda lebih suka menggunakan repositori Git atau Subversion, jangan gunakan dari cabang master. Sebagai gantinya, beri tag pada versi yang disetujui dan tentukan versi tersebut sebagai sumber aplikasi atau buku masak.

Menerapkan Kode ke Instans Online

AWS OpsWorks Tumpukan tidak secara otomatis menyebarkan kode yang diperbarui ke instance online. Anda harus melakukan operasi itu secara manual, yang menimbulkan tantangan berikut:

- Menyebarkan pembaruan secara efisien tanpa mengorbankan kemampuan situs untuk menangani permintaan pelanggan selama proses penyebaran.
- Menangani penerapan yang gagal, baik karena masalah dengan aplikasi atau buku masak yang diterapkan atau masalah dengan proses penerapan itu sendiri.

Pendekatan paling sederhana adalah menjalankan perintah Deploy default (untuk aplikasi) atau Perbarui perintah Cookbooks Kustom (untuk buku masak), yang menyebarkan pembaruan ke setiap instance secara bersamaan. Pendekatan ini sederhana dan cepat, tetapi tidak ada margin untuk

kesalahan. Jika penerapan gagal atau kode yang diperbarui memiliki masalah, setiap instance di tumpukan produksi Anda dapat terpengaruh, berpotensi mengganggu atau menonaktifkan situs Anda hingga Anda dapat memperbaiki masalah atau memutar kembali ke versi sebelumnya.

Rekomendasi: Gunakan strategi penerapan yang kuat, yang memungkinkan instance yang menjalankan kode versi lama untuk terus menangani permintaan hingga Anda memverifikasi bahwa penerapan berhasil dan dengan percaya diri dapat mentransfer semua lalu lintas masuk ke versi baru.

Bagian berikut memberikan dua contoh strategi penyebaran yang kuat, diikuti dengan diskusi tentang cara mengelola database backend selama penerapan. Untuk singkatnya, mereka menjelaskan pembaruan aplikasi, tetapi Anda dapat menggunakan strategi serupa untuk buku masak.

Topik

- Menggunakan Rolling Deployment
- Menggunakan Tumpukan Terpisah
- Mengelola Database Backend

Menggunakan Rolling Deployment

Penerapan bergulir memperbarui aplikasi pada instance server aplikasi online stack dalam beberapa fase. Dengan setiap fase, Anda memperbarui subset instans online dan memverifikasi bahwa pembaruan berhasil sebelum memulai fase berikutnya. Jika Anda mengalami masalah, instance yang masih menjalankan versi aplikasi lama dapat terus menangani lalu lintas masuk hingga Anda menyelesaikan masalah.

Contoh berikut mengasumsikan bahwa Anda menggunakan praktik yang disarankan untuk mendistribusikan instance server aplikasi stack Anda di beberapa Availability Zone.

Untuk melakukan penyebaran bergulir

1. Pada halaman Deploy App, pilih Advanced, pilih satu instance server aplikasi, dan deploy aplikasi ke instance tersebut.

   Jika ingin berhati-hati, Anda dapat menghapus instance dari penyeimbang beban sebelum menerapkan aplikasi. Ini memastikan bahwa pengguna tidak akan menemukan aplikasi yang diperbarui sampai Anda memverifikasi bahwa itu berfungsi dengan benar. Jika Anda menggunakan Elastic Load Balancing, hapus instance dari load balancer dengan menggunakan konsol Elastic Load Balancing, CLI, atau SDK.

2. Pastikan aplikasi yang diperbarui berfungsi dengan benar dan instans memiliki metrik kinerja yang dapat diterima.

   Jika Anda menghapus instance dari penyeimbang beban Elastic Load Balancing, gunakan konsol Elastic Load Balancing, CLI, atau SDK untuk memulihkannya. Versi aplikasi yang diperbarui sekarang menangani permintaan pengguna.
3. Terapkan pembaruan ke sisa instance di Availability Zone dan verifikasi bahwa mereka berfungsi dengan benar dan memiliki metrik yang dapat diterima.
4. Ulangi langkah 3 untuk Availability Zone stack lainnya, satu zona pada satu waktu. Jika Anda ingin sangat berhati-hati, ulangi langkah 1 — 3.

> **Note**
>
> Jika Anda menggunakan penyeimbang beban Elastic Load Balancing, Anda dapat menggunakan pemeriksaan kesehatannya untuk memverifikasi bahwa penerapan berhasil. Namun, atur jalur ping ke aplikasi yang memeriksa dependensi dan memverifikasi bahwa semuanya berfungsi dengan benar, bukan file statis yang hanya mengonfirmasi bahwa server aplikasi sedang berjalan.

Menggunakan Tumpukan Terpisah

Pendekatan lain untuk mengelola aplikasi adalah dengan menggunakan tumpukan terpisah untuk setiap fase siklus hidup aplikasi. Tumpukan yang berbeda kadang-kadang disebut sebagai lingkungan. Pengaturan ini memungkinkan Anda untuk melakukan pengembangan dan pengujian pada tumpukan yang tidak dapat diakses publik. Saat Anda siap untuk menerapkan pembaruan, alihkan lalu lintas pengguna dari tumpukan yang menghosting versi aplikasi saat ini ke tumpukan yang menghosting versi yang diperbarui.

Topik

- Menggunakan Tumpukan Pengembangan, Pementasan, dan Produksi
- Menggunakan Strategi Penerapan Biru-Hijau

Menggunakan Tumpukan Pengembangan, Pementasan, dan Produksi

Pendekatan yang paling umum menggunakan tumpukan berikut.

Tumpukan Pengembangan

Gunakan tumpukan pengembangan untuk tugas-tugas seperti menerapkan fitur baru atau memperbaiki bug. Tumpukan pengembangan pada dasarnya adalah tumpukan produksi prototipe, dengan lapisan, aplikasi, sumber daya, dan sebagainya yang sama yang disertakan pada tumpukan produksi Anda. Karena tumpukan pengembangan biasanya tidak harus menangani beban yang sama dengan tumpukan produksi, Anda biasanya dapat menggunakan instance yang lebih sedikit atau lebih kecil.

Tumpukan pengembangan tidak dihadapi publik; Anda mengontrol akses sebagai berikut:

- Batasi akses jaringan dengan mengonfigurasi aturan masuk grup keamanan server aplikasi atau load balancer untuk menerima permintaan masuk hanya dari alamat IP atau rentang alamat tertentu.

  Misalnya, batasi akses HTTP, HTTPS, dan SSH ke alamat dalam rentang alamat perusahaan Anda.
- Kontrol akses ke fungsionalitas manajemen tumpukan AWS OpsWorks Stacks dengan menggunakan halaman Izin tumpukan.

  Misalnya, berikan tingkat izin Kelola ke tim pengembangan, dan Tampilkan izin kepada semua karyawan lainnya.

Stack Pementasan

Gunakan staging stack untuk menguji dan menyelesaikan kandidat untuk tumpukan produksi yang diperbarui. Ketika Anda telah menyelesaikan pengembangan, buat staging stack dengan mengkloning tumpukan pengembangan. Kemudian jalankan rangkaian pengujian Anda di staging stack dan terapkan pembaruan ke tumpukan itu untuk memperbaiki masalah yang muncul.

Stack staging juga tidak menghadap publik; Anda mengontrol tumpukan dan akses jaringan dengan cara yang sama seperti yang Anda lakukan untuk tumpukan pengembangan. Perhatikan bahwa saat Anda mengkloning tumpukan pengembangan untuk membuat staging stack, Anda dapat mengkloning izin yang diberikan oleh AWS OpsWorks manajemen izin Stacks. Namun, kloning tidak memengaruhi izin yang diberikan oleh kebijakan IAM pengguna. Anda harus menggunakan konsol IAM, CLI, atau SDK untuk mengubah izin tersebut. Untuk informasi selengkapnya, lihat Mengelola izin.

Tumpukan Produksi

Tumpukan produksi adalah tumpukan yang menghadap publik yang mendukung aplikasi Anda saat ini. Ketika staging stack telah lulus pengujian, Anda mempromosikannya ke produksi dan

menghentikan tumpukan produksi lama. Untuk contoh cara melakukannya, lihat Menggunakan Strategi Penerapan Biru-Hijau.

> **Note**
>
> Alih-alih menggunakan konsol AWS OpsWorks Stacks untuk membuat tumpukan secara manual, buat AWS CloudFormation template untuk setiap tumpukan. Pendekatan ini memiliki keuntungan sebagai berikut:
>
> - Kecepatan dan kenyamanan — Saat Anda meluncurkan template, AWS CloudFormation secara otomatis membuat tumpukan, termasuk semua instance yang diperlukan.
> - Konsistensi — Simpan template untuk setiap tumpukan di repositori sumber Anda untuk memastikan bahwa pengembang menggunakan tumpukan yang sama untuk tujuan yang sama.

Menggunakan Strategi Penerapan Biru-Hijau

Strategi penyebaran biru-hijau adalah salah satu cara umum untuk menggunakan tumpukan terpisah secara efisien untuk menyebarkan pembaruan aplikasi ke produksi.

- Lingkungan biru adalah tumpukan produksi, yang menampung aplikasi saat ini.
- Lingkungan hijau adalah staging stack, yang menampung aplikasi yang diperbarui.

Saat Anda siap untuk menerapkan aplikasi yang diperbarui ke produksi, Anda mengalihkan lalu lintas pengguna dari tumpukan biru ke tumpukan hijau, yang menjadi tumpukan produksi baru. Anda kemudian pensiun dari tumpukan biru tua.

Contoh berikut menjelaskan cara melakukan penyebaran biru-hijau dengan AWS OpsWorks tumpukan Stacks, bersama dengan Route 53 dan kumpulan penyeimbang beban Elastic Load Balancing. Sebelum beralih, Anda harus memastikan hal berikut:

- Pembaruan aplikasi pada tumpukan hijau telah lulus pengujian dan siap untuk produksi.
- Tumpukan hijau identik dengan tumpukan biru kecuali bahwa itu termasuk aplikasi yang diperbarui dan tidak menghadap publik.

  Kedua tumpukan memiliki izin yang sama, jumlah dan jenis instance yang sama di setiap lapisan, konfigurasi berbasis waktu dan beban yang sama, dan sebagainya.

- Semua instance 24/7 tumpukan hijau dan instance berbasis waktu terjadwal sedang online.
- Anda memiliki kumpulan penyeimbang beban Elastic Load Balancing yang dapat dipasang secara dinamis ke lapisan di kedua tumpukan dan dapat dipanaskan terlebih dahulu untuk menangani volume lalu lintas yang diharapkan.
- Anda telah menggunakan fitur routing tertimbang Route 53 untuk membuat catatan yang ditetapkan di zona yang dihosting yang menyertakan penyeimbang beban gabungan Anda.
- Anda telah menetapkan bobot bukan nol ke penyeimbang beban yang dilampirkan ke lapisan server aplikasi tumpukan biru Anda dan bobot nol ke penyeimbang beban yang tidak digunakan. Ini memastikan bahwa penyeimbang beban tumpukan biru menangani semua lalu lintas yang masuk.

Untuk mengalihkan pengguna ke tumpukan hijau

1. Lampirkan salah satu load balancer pool yang tidak digunakan ke layer server aplikasi tumpukan hijau. Dalam beberapa skenario, seperti ketika Anda mengharapkan lalu lintas kilat, atau jika Anda tidak dapat mengonfigurasi uji beban untuk meningkatkan lalu lintas secara bertahap, pra-hangatkan penyeimbang beban untuk menangani lalu lintas yang diharapkan.
2. Setelah semua instance tumpukan hijau telah lulus pemeriksaan kesehatan Elastic Load Balancing, ubah bobot dalam set rekor Route 53 sehingga penyeimbang beban tumpukan hijau memiliki bobot bukan nol dan penyeimbang beban tumpukan biru memiliki bobot yang berkurang. Kami menyarankan Anda memulai dengan meminta tumpukan hijau menangani sebagian kecil permintaan, mungkin 5%, dengan tumpukan biru menangani sisanya. Anda sekarang memiliki dua tumpukan produksi, dengan tumpukan hijau menangani beberapa permintaan yang masuk dan tumpukan biru menangani sisanya.
3. Pantau metrik kinerja tumpukan hijau. Jika mereka dapat diterima, tingkatkan bobot tumpukan hijau sehingga menangani mungkin 10% dari lalu lintas yang masuk.
4. Ulangi Langkah 3 sampai tumpukan hijau menangani sekitar setengah dari lalu lintas yang masuk. Masalah apa pun seharusnya muncul pada titik ini, jadi jika tumpukan hijau berkinerja baik, Anda dapat menyelesaikan prosesnya dengan mengurangi bobot tumpukan biru menjadi nol. Tumpukan hijau sekarang menjadi tumpukan biru baru dan menangani semua lalu lintas yang masuk.
5. Lepaskan penyeimbang beban dari layer server aplikasi tumpukan biru tua dan kembalikan ke kolam.
6. Meskipun tumpukan biru lama tidak lagi menangani permintaan pengguna, kami sarankan untuk menyimpannya untuk sementara waktu jika ada masalah dengan tumpukan biru baru. Dalam hal ini, Anda dapat memutar kembali pembaruan dengan membalikkan prosedur untuk

mengarahkan lalu lintas masuk kembali ke tumpukan biru lama. Ketika Anda yakin bahwa tumpukan biru baru beroperasi dengan baik, matikan tumpukan biru lama.

Mengelola Database Backend

Jika aplikasi Anda bergantung pada basis data backend, Anda harus beralih dari aplikasi lama ke yang baru. AWS OpsWorks Stacks mendukung opsi database berikut.

Lapisan Amazon RDS

Dengan lapisan Amazon Relational Database Service (Amazon RDS), Anda membuat instans RDS DB secara terpisah dan kemudian mendaftarkannya ke tumpukan Anda. Anda dapat mendaftarkan instans RDS DB dengan hanya satu tumpukan pada satu waktu, tetapi Anda dapat mengganti instance RDS DB dari satu tumpukan ke tumpukan lainnya.

AWS OpsWorks Stacks menginstal file dengan data koneksi pada server aplikasi Anda dalam format yang mudah dapat digunakan oleh aplikasi Anda. AWS OpsWorks Stacks juga menambahkan informasi koneksi database ke konfigurasi stack dan atribut deployment, yang dapat diakses oleh resep. Anda juga dapat menggunakan JSON untuk menyediakan data koneksi ke aplikasi. Untuk informasi selengkapnya, lihat Menghubungkan ke Database.

Memperbarui aplikasi yang bergantung pada database menimbulkan dua tantangan dasar:

- Memastikan bahwa setiap transaksi dicatat dengan benar selama transisi sambil juga menghindari kondisi balapan antara versi aplikasi baru dan lama.
- Melakukan transisi dengan cara yang membatasi dampak pada kinerja situs Anda dan meminimalkan atau menghilangkan downtime.

Saat Anda menggunakan strategi penyebaran yang dijelaskan dalam topik ini, Anda tidak bisa begitu saja melepaskan database dari aplikasi lama dan memasangnya kembali ke yang baru. Kedua versi aplikasi berjalan secara paralel selama transisi dan harus memiliki akses ke data yang sama. Berikut ini menjelaskan dua pendekatan untuk mengelola transisi, yang keduanya memiliki kelebihan dan tantangan.

Pendekatan 1: Minta kedua aplikasi terhubung ke database yang sama

Keuntungan

- Tidak ada downtime selama transisi.

Satu aplikasi secara bertahap berhenti mengakses database sementara yang lain secara bertahap mengambil alih.

- Anda tidak perlu menyinkronkan data antara dua database.

Tantangan

- Kedua aplikasi mengakses database yang sama, jadi Anda harus mengelola akses untuk mencegah kehilangan data atau korupsi.
- Jika Anda perlu bermigrasi ke skema database baru, versi aplikasi lama harus dapat menggunakan skema baru.

Jika Anda menggunakan tumpukan terpisah, pendekatan ini mungkin paling cocok untuk Amazon RDS karena instance tidak terikat secara permanen ke tumpukan tertentu dan dapat diakses oleh aplikasi yang berjalan pada tumpukan yang berbeda. Namun, Anda tidak dapat mendaftarkan instans RDS DB dengan lebih dari satu tumpukan sekaligus, jadi Anda harus memberikan data koneksi ke kedua aplikasi, misalnya dengan menggunakan JSON. Untuk informasi selengkapnya, lihat Menggunakan Resep Kustom.

Jika Anda menggunakan upgrade bergulir, versi aplikasi lama dan baru di-host pada tumpukan yang sama, sehingga Anda dapat menggunakan lapisan Amazon RDS atau MySQL.

Pendekatan 2: Menyediakan setiap versi aplikasi dengan basis datanya sendiri

Keuntungan

- Setiap versi memiliki database sendiri, sehingga skema tidak harus kompatibel.

Tantangan

- Sinkronisasi data antara dua database selama transisi tanpa kehilangan atau merusak data.
- Memastikan bahwa prosedur sinkronisasi Anda tidak menyebabkan downtime yang signifikan atau secara signifikan menurunkan kinerja situs.

Jika Anda menggunakan tumpukan terpisah, masing-masing memiliki database sendiri. Jika Anda menggunakan penerapan bergulir, Anda dapat melampirkan dua database ke tumpukan, satu untuk setiap aplikasi. Jika aplikasi lama dan yang diperbarui tidak memiliki skema database yang kompatibel, pendekatan ini lebih baik.

Rekomendasi: Secara umum, sebaiknya gunakan lapisan Amazon RDS sebagai basis data backend aplikasi karena lebih fleksibel dan dapat digunakan untuk skenario transisi apa pun. Untuk informasi selengkapnya tentang cara menangani transisi, lihat Panduan Pengguna Amazon RDS.

Ketergantungan Buku Masak Kemasan Secara Lokal

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menggunakan Berkshelf untuk mengemas dependensi buku masak Anda secara lokal, mengunggah paket ke Amazon S3, dan memodifikasi tumpukan Anda untuk menggunakan paket di Amazon S3 sebagai sumber buku masak. Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan bucket S3? atau Bagaimana Cara Menghapus S3? .

Panduan berikut menjelaskan cara mengemas buku masak Anda dan dependensinya ke file .zip, lalu menggunakan file .zip sebagai sumber buku masak Anda untuk instance Linux di Stacks. AWS OpsWorks Panduan pertama menjelaskan cara mengemas satu buku masak. Panduan kedua menjelaskan cara mengemas beberapa buku masak.

Sebelum Anda mulai, instal Chef Development Kit (juga dikenal sebagai Chef DK), yang merupakan bermacam-macam alat yang dibangun oleh komunitas Chef. Anda akan membutuhkan ini untuk menggunakan alat `chef` baris perintah.

Dependensi Pengemasan Lokal di Chef 12

Di Chef 12 Linux, Berkshelf tidak lagi diinstal secara default pada instance stack. Kami menyarankan Anda menginstal dan menggunakan Berkshelf pada komputer pengembangan lokal untuk mengemas dependensi buku masak Anda secara lokal. Unggah paket Anda, dengan dependensi yang disertakan, ke Amazon S3. Terakhir, ubah tumpukan Chef 12 Linux Anda untuk menggunakan paket yang diunggah sebagai sumber buku masak. Waspadai perbedaan berikut saat Anda mengemas buku masak di Chef 12.

1. Di komputer lokal, buat buku masak dengan menjalankan alat baris `chef` perintah.

   ```
   chef generate cookbook "server-app"
   ```

Perintah ini membuat buku masak, Berksfile, `metadata.rb` file, dan direktori resep, dan menempatkannya di folder yang memiliki nama yang sama dengan buku masak. Contoh berikut menunjukkan struktur apa yang dibuat.

```
server-app <-- the cookbook you've just created
    ### Berksfile
    ### metadata.rb
    ### recipes
```

2. Dalam editor teks, edit Berksfile untuk menunjuk ke buku masak di mana buku masak akan `server-app` bergantung. Dalam contoh kami, kami `server-app` ingin bergantung pada `java`buku masak dari Chef Supermarket. Kami menentukan versi 1.50.0 atau versi minor yang lebih baru, tetapi Anda dapat memasukkan versi yang diterbitkan dalam tanda kutip tunggal. Simpan perubahan Anda dan tutup file.

```
source 'https://supermarket.chef.io'
cookbook 'java', '~> 1.50.0'
```

3. Edit `metadata.rb` file untuk menambahkan ketergantungan. Simpan perubahan Anda dan tutup file.

```
depends 'java' , '~> 1.50.0'
```

4. Ubah ke direktori `server-app` buku masak yang dibuat Chef untuk Anda, lalu jalankan `package` perintah untuk membuat `tar` file buku masak. Jika Anda mengemas beberapa buku masak, Anda ingin menjalankan perintah ini di direktori root tempat semua buku masak disimpan. Untuk mengemas satu buku masak, jalankan perintah ini di tingkat direktori buku masak. Dalam contoh ini, kami menjalankan perintah ini di `server-app` direktori.

```
berks package cookbooks.tar.gz
```

Output-nya menyerupai yang berikut. `tar.gz`File dibuat di direktori lokal Anda.

```
Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
```

5. Di AWS CLI, unggah paket yang baru saja Anda buat ke Amazon S3. Catat URL baru paket buku masak setelah Anda mengunggahnya ke S3; Anda memerlukan URL ini untuk pengaturan tumpukan Anda.

```
aws s3 cp cookbooks.tar.gz s3://bucket-name/
```

Output-nya menyerupai yang berikut.

```
upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
```

6. Di AWS OpsWorks Stacks, ubah tumpukan Anda untuk menggunakan paket yang Anda unggah sebagai sumber buku masak.

   a. Setel pengaturan Gunakan buku masak Chef kustom ke Ya.

   b. Setel jenis Repositori ke Arsip S3.

   c. Di URL Repositori, tempel URL paket buku masak yang Anda unggah di langkah 5.

   Simpan perubahan tumpukan Anda.

Dependensi Kemasan Secara Lokal untuk Satu Buku Masak

1. Di komputer lokal, buat buku masak dengan menggunakan alat baris perintah koki:

   ```
   chef generate cookbook "server-app"
   ```

   Perintah ini membuat buku masak dan Berksfile, dan menempatkannya di folder yang memiliki nama yang sama dengan buku masak.

2. Ubah ke direktori buku masak yang dibuat Chef untuk Anda, dan kemudian paket semuanya dengan menjalankan perintah berikut:

   ```
   berks package cookbooks.tar.gz
   ```

   Keluaran terlihat seperti ini:

   ```
   Cookbook(s) packaged to /Users/username/tmp/berks/cookbooks.tar.gz
   ```

3. Di AWS CLI, unggah paket yang baru saja Anda buat ke Amazon S3:

   ```
   aws s3 cp cookbooks.tar.gz s3://bucket-name/
   ```

Keluaran terlihat seperti ini:

```
upload: ./cookbooks.tar.gz to s3://bucket-name/cookbooks.tar.gz
```

4. Di AWS OpsWorks Stacks, ubah tumpukan Anda untuk menggunakan paket yang Anda unggah sebagai sumber buku masak.

Dependensi Kemasan Secara Lokal untuk Beberapa Buku Masak

Contoh ini membuat dua buku masak dan mengemas dependensi untuk mereka.

1. Di komputer lokal, jalankan `chef` perintah berikut untuk menghasilkan dua buku masak:

```
chef generate cookbook "server-app"
chef generate cookbook "server-utils"
```

Dalam contoh ini, buku masak server-app melakukan konfigurasi Java, jadi kita perlu menambahkan ketergantungan pada Java.

2. Edit `server-app/metadata.rb` untuk menambahkan ketergantungan pada buku masak Java komunitas:

```
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. Beri tahu Berkshelf apa yang harus dikemas dengan mengedit file Berksfile di direktori root buku masak sebagai berikut:

```
source "https://supermarket.chef.io"
cookbook "server-app", path: "./server-app"
cookbook "server-utils", path: "./server-utils"
```

Struktur file Anda sekarang terlihat seperti ini:

```
 ..
```

```
    ### Berksfile
    ### server-app
    ### server-utils
```

4. Terakhir, buat paket zip, unggah ke Amazon S3, dan ubah tumpukan AWS OpsWorks Stacks Anda untuk menggunakan sumber buku masak baru. Untuk melakukan ini, ikuti langkah 2 hingga 4 inDependensi Kemasan Secara Lokal untuk Satu Buku Masak.

Sumber daya tambahan

Untuk informasi selengkapnya tentang dependensi buku masak kemasan, lihat berikut ini.

- Cara Mengemas Dependensi Cookbook Secara Lokal dengan Berkshelf di Blog AWS DevOps
- Linux Chef 12 dengan Berkshelf di forum AWS OpsWorks
- Berkshelf di Chef 12 di forum AWS OpsWorks
- Menginstal Buku Masak Kustom dalam panduan ini
- Repositori Buku Masak dalam panduan ini

Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Stack adalah entitas AWS OpsWorks Stacks tingkat atas. Ini merupakan serangkaian instans yang ingin Anda kelola secara kolektif, biasanya karena instans memiliki tujuan umum seperti melayani aplikasi PHP. Selain berfungsi sebagai kontainer, tumpukan menangani tugas-tugas yang berlaku untuk grup instans secara keseluruhan, seperti mengelola aplikasi dan buku masak.

Misalnya, tumpukan yang tujuannya adalah untuk melayani aplikasi web mungkin terlihat seperti berikut:

- Satu set instance server aplikasi, yang masing-masing menangani sebagian dari lalu lintas yang masuk.
- Load balancer instance, yang mengambil lalu lintas masuk dan mendistribusikannya ke seluruh server aplikasi.
- Sebuah instance database, yang berfungsi sebagai penyimpanan data back-end untuk server aplikasi.

Praktik yang umum adalah memiliki banyak tumpukan yang mewakili lingkungan yang berbeda. Satu set tumpukan khas terdiri dari:

- Tumpukan pengembangan yang akan digunakan oleh pengembang untuk menambahkan fitur, memperbaiki bug, dan melakukan tugas pengembangan dan pemeliharaan lainnya.
- Tumpukan pementasan untuk memverifikasi pembaruan atau perbaikan sebelum mengeksposnya secara publik.
- Tumpukan produksi, yang merupakan versi publik yang menangani permintaan masuk dari pengguna.

Bagian ini menjelaskan dasar-dasar bekerja dengan tumpukan.

Topik

- Memindahkan tumpukan dari Amazon EC2 -Classic ke VPC
- Membuat Tumpukan Baru
- Menjalankan Stack di VPC
- Perbarui Tumpukan
- Klon Tumpukan
- Jalankan AWS OpsWorks Stacks Stack Commands
- Menggunakan JSON Kustom
- Hapus Tumpukan

Memindahkan tumpukan dari Amazon EC2 -Classic ke VPC

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini menjelaskan cara memigrasikan AWS OpsWorks Stacks tumpukan dari platform jaringan Amazon EC2 Classic ke jaringan Amazon Virtual Private Cloud (Amazon VPC).

Jika Anda membuat AWS akun sebelum 04-12-2013, Anda mungkin memiliki dukungan untuk EC2 -Classic di sebagian Wilayah. AWS Beberapa EC2 sumber daya dan fitur Amazon, seperti peningkatan jaringan dan tipe instans yang baru, memerlukan virtual private cloud (VPC). Beberapa sumber daya dapat dibagi ke VPC EC2 dan VPC, tetapi sebagian tidak dapat dibagi. Untuk menghindari gangguan pada layanan Anda, sebaiknya Anda memigrasikan AWS OpsWorks Stacks tumpukan ke VPC.

Topik

- Prasyarat
- Migrasi AWS OpsWorks Stacks tumpukan ke VPC
- Lihat juga

Prasyarat

Sebelum memulai, Anda harus memiliki VPC yang memenuhi persyaratan AWS OpsWorks Stacks konfigurasi. Untuk mengonfigurasi subnet pribadi di VPC AWS OpsWorks Stacks Anda, Menjalankan Stack di VPC lihat di panduan ini. Anda dapat membuat VPC khusus dengan menggunakan konsol manajemen Amazon VPC. Untuk informasi selengkapnya, lihat konfigurasi VPCs dan subnet wizard konsol VPC Amazon VPC di Panduan Pengguna Amazon Virtual Private Cloud.

Untuk melanjutkan migrasi, Anda memerlukan ID VPC dan ID subnet yang ingin Anda gunakan.

Migrasi AWS OpsWorks Stacks tumpukan ke VPC

Pertama, kloning tumpukan EC2 -Classic yang ada dengan menggunakan AWS OpsWorks Stacks konsol atau API. Kemudian, pindahkan sumber daya tumpukan yang ada ke tumpukan baru. Mulai instance baru di tumpukan kloning, dan terapkan aplikasi. Pastikan tumpukan baru berfungsi. Terakhir, hapus sumber daya EC2 -Classic dari tumpukan EC2 -Classic, lalu hapus tumpukan lama.

1. Kloning tumpukan EC2 -Classic Anda yang ada ke VPC Anda. Mengkloning tumpukan menyalin pengaturan tumpukan, lapisan, aplikasi, pengguna, dan izin pengguna ke tumpukan baru. Untuk informasi lebih lanjut tentang cara mengkloning tumpukan, lihat Klon Tumpukan di panduan ini.

   Anda juga dapat mengkloning tumpukan dengan menggunakan AWS OpsWorks Stacks API. Saat Anda mengkloning tumpukan dengan menggunakan AWS CLI or AWS SDKs, atur nilai `VpcId` parameter ke ID VPC yang Anda buat. Prasyarat Untuk informasi selengkapnya, lihat CloneStack di dalam Referensi API AWS OpsWorks Stacks .
2. Buat instance baru di lapisan tumpukan kloning. Pastikan untuk menentukan ID subnet yang Anda buat. Prasyarat Untuk informasi selengkapnya tentang cara membuat instans di tumpukan, lihat Menambahkan Instance ke Layer dalam panduan ini.
3. Migrasikan sumber daya klasik Anda, seperti grup EC2 keamanan, penyeimbang beban Elastic Load Balancing, dan alamat IP Elastis ke VPC Anda, lalu kaitkan dengan tumpukan kloning. Untuk informasi selengkapnya, lihat Memigrasi sumber daya Anda ke VPC di Panduan Pengguna EC2 Amazon.
4. Daftarkan volume Amazon EBS dan instans Amazon RDS dengan tumpukan kloning. Untuk informasi selengkapnya tentang cara mendaftarkan sumber daya dengan tumpukan, lihat Mendaftarkan Sumber Daya dengan Stack dalam panduan ini.

   Volume Amazon EBS tidak terkait dengan VPC, dan Anda dapat menggunakannya di seluruh instance di EC2 kedua tumpukan dan tumpukan -Classic di VPC. Anda dapat mendaftarkan instans Amazon RDS di EC2 -Classic dengan tumpukan -Classic dan EC2 tumpukan di VPC.
5. Mulai instance di tumpukan kloning, lalu pindahkan sebagian kecil beban kerja Anda ke tumpukan kloning. Misalnya, pindahkan sebagian kecil lalu lintas ke penyeimbang beban Elastic Load Balancing di tumpukan kloning. Jika Anda menggunakan Amazon Route 53, lihat Merutekan lalu lintas ke penyeimbang beban ELB di Panduan Pengembang Amazon Route 53.

   Rutekan hanya sebagian kecil lalu lintas sampai Anda yakin bahwa tumpukan baru berfungsi dan mendukung aplikasi Anda. Biarkan tumpukan baru bekerja dengan persentase kecil lalu

lintas untuk masa percobaan, seperti seminggu. Setelah Anda memverifikasi bahwa tumpukan baru berfungsi, rute lalu lintas yang tersisa ke tumpukan.

6. Setelah Anda yakin tumpukan kloning berfungsi, pindahkan sisa trafik produksi atau beban kerja Anda ke tumpukan kloning. Anda sekarang dapat menghentikan instance di tumpukan EC2 -Classic. Kami menyarankan agar tumpukan lama tetap tersedia selama beberapa minggu, sehingga Anda dapat memindahkan beban kerja kembali ke tumpukan lama jika ada masalah yang terjadi dengan tumpukan baru dalam beberapa minggu setelah migrasi.
7. Ketika tumpukan baru telah berfungsi selama beberapa minggu, hapus instance di tumpukan EC2 -Classic. Untuk informasi lebih lanjut tentang cara menghapus instans, lihat Menghapus Instans AWS OpsWorks Stacks di panduan ini.

   > **Important**
   >
   > Jangan gunakan EC2 konsol Amazon atau API untuk menghentikan atau menghapus AWS OpsWorks instance.

8. Hapus aplikasi di tumpukan EC2 -Classic. Untuk informasi selengkapnya tentang cara menghapus aplikasi, lihat Untuk menghapus aplikasi dari tumpukan dalam panduan ini.
9. Hapus tumpukan EC2 -Classic. Untuk informasi lebih lanjut tentang cara menghapus tumpukan, lihat Hapus Tumpukan di panduan ini.

Lihat juga

- Memindahkan dari EC2 -Classic ke VPC
- Panduan Debugging dan Pemecahan Masalah
- Menjalankan Stack di VPC

Membuat Tumpukan Baru

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk membuat tumpukan baru, di dasbor AWS OpsWorks Stacks, klik Tambah tumpukan. Anda kemudian dapat menggunakan halaman Add Stack untuk mengkonfigurasi tumpukan. Setelah selesai, klik Tambahkan Tumpukan.

Topik

- Pilih Jenis Stack untuk Membuat
- Opsi Basic
- Opsi Lanjutan

Pilih Jenis Stack untuk Membuat

Sebelum Anda membuat tumpukan, Anda harus memutuskan jenis tumpukan yang ingin Anda buat. Untuk bantuan, lihat tabel berikut ini.

Jika Anda ingin membuat...	Buat jenis tumpukan ini jika Anda ingin...	Untuk mempelajari caranya, ikuti petunjuk ini:
Tumpukan sampel	Jelajahi dasar-dasar AWS OpsWorks dengan tumpukan Chef 12 berbasis Linux dan contoh aplikasi Node.js.	Memulai: Contoh
Tumpukan Chef 12 berbasis Linux	Buat tumpukan berbasis Linux yang menggunakan Chef versi terbaru yang didukung AWS. OpsWorks Pilih opsi ini jika Anda adalah pengguna Chef tingkat lanjut yang ingin mendapatkan manfaat dari banyak pilihan buku masak komunitas atau menulis buku masak khusus Anda sendiri.	Memulai: Linux

Jika Anda ingin membuat...	Buat jenis tumpukan ini jika Anda ingin...	Untuk mempelajari caranya, ikuti petunjuk ini:
	Untuk informasi selengkapnya, lihat Koki 12 Linux.	
Tumpukan Chef 12.2 berbasis Windows	Buat tumpukan berbasis Windows.	Memulai: Windows
Tumpukan Chef 11.10 berbasis Linux	Buat tumpukan ini jika organisasi Anda memerlukan penggunaan Chef 11.10 dengan Linux untuk kompatibilitas mundur.	Memulai dengan Chef 11 Linux Stacks

Opsi Basic

Halaman Add Stack memiliki opsi dasar berikut.

Nama tumpukan

(Wajib) Nama yang digunakan untuk mengidentifikasi tumpukan di konsol AWS OpsWorks Stacks. Nama tidak harus unik. AWS OpsWorks Stacks juga menghasilkan ID tumpukan, yang merupakan GUID yang secara unik mengidentifikasi tumpukan. Misalnya, dengan perintah AWS CLI seperti update-stack, Anda menggunakan ID tumpukan untuk mengidentifikasi tumpukan tertentu. Setelah Anda membuat tumpukan, Anda dapat menemukan IDnya dengan memilih Stack di panel navigasi dan kemudian memilih Pengaturan Stack. ID tersebut diberi label OpsWorks ID.

Wilayah

(Wajib) Wilayah AWS tempat instans akan diluncurkan.

VPC

(Opsional) ID VPC tempat tumpukan akan diluncurkan. Semua instance akan diluncurkan ke VPC ini, dan Anda tidak dapat mengubah ID nanti.

- Jika akun Anda mendukung EC2 Classic, Anda dapat menentukan No VPC (nilai default) jika Anda tidak ingin menggunakan VPC.

  Untuk informasi selengkapnya tentang EC2 Classic, lihat Platform yang Didukung.

- Jika akun Anda tidak mendukung EC2 Classic, Anda harus menentukan VPC.

  Pengaturan default adalah VPC Default, yang menggabungkan kemudahan penggunaan EC2 Classic dengan manfaat fitur jaringan VPC. Jika Anda ingin menjalankan tumpukan Anda di VPC biasa, Anda harus membuatnya dengan menggunakan konsol VPC, API, atau CLI. Untuk informasi selengkapnya tentang cara membuat VPC untuk tumpukan AWS OpsWorks Stacks, lihat. Menjalankan Stack di VPC Untuk informasi umum, lihat Amazon Virtual Private Cloud.

Zona Ketersediaan Default/Subnet default

(Opsional) Pengaturan ini tergantung pada apakah Anda membuat tumpukan Anda di VPC:

- Jika akun Anda mendukung EC2 Classic dan Anda menyetel VPC ke No VPC, pengaturan ini diberi label Default Availability Zone, yang menentukan AWS Availability Zone default tempat instance akan diluncurkan.
- Jika akun Anda tidak mendukung EC2 Classic atau Anda memilih untuk menentukan VPC, bidang ini diberi label Default subnet, yang menentukan subnet default tempat instance akan diluncurkan. Anda dapat meluncurkan instance di subnet lain dengan mengganti nilai ini saat membuat instance. Setiap subnet dikaitkan dengan satu Availability Zone.

Anda dapat meminta AWS OpsWorks Stacks meluncurkan instance di Availability Zone atau subnet yang berbeda dengan mengganti setelan ini saat Anda membuat instance.

Untuk informasi lebih lanjut tentang cara menjalankan tumpukan di VPC, lihat. Menjalankan Stack di VPC

Sistem operasi default

(Opsional) Sistem operasi yang diinstal secara default pada setiap instans. Anda memiliki opsi berikut:

- Salah satu sistem operasi Linux bawaan.
- Microsoft Windows Server 2012 R2.
- AMI khusus berdasarkan salah satu sistem operasi yang didukung.

  Jika Anda memilih Gunakan AMI kustom, sistem operasi ditentukan oleh AMI kustom yang Anda tentukan saat membuat instance. Untuk informasi selengkapnya, lihat Menggunakan Custom AMIs.

Untuk informasi lebih lanjut tentang sistem operasi yang tersedia, lihatAWS OpsWorks Stacks sistem operasi.

> **Note**
>
> Anda dapat mengganti sistem operasi default saat Anda membuat instans. Namun, Anda tidak dapat mengganti sistem operasi Linux untuk menentukan Windows, atau Windows untuk menentukan sistem operasi Linux.

Kunci SSH default

(Opsional) Amazon EC2 key pair dari wilayah stack. Nilai default-nya tidak ada. Jika Anda menentukan key pair, AWS OpsWorks Stacks akan menginstal public key pada instance.

- Dengan instans Linux, Anda dapat menggunakan kunci privat dengan klien SSH untuk login ke instans tumpukan.

  Untuk informasi selengkapnya, lihat Login dengan SSH.
- Dengan instance Windows, Anda dapat menggunakan kunci pribadi dengan EC2 konsol Amazon atau CLI untuk mengambil kata sandi Administrator instans.

  Anda kemudian dapat menggunakan kata sandi itu dengan klien RDP untuk masuk ke instance sebagai Administrator. Untuk informasi selengkapnya, lihat Masuk dengan RDP.

Untuk informasi lebih lanjut tentang cara mengelola kunci SSH, lihatMengelola Akses SSH.

> **Note**
>
> Anda dapat mengganti pengaturan ini dengan menentukan pasangan kunci yang berbeda, atau tidak ada pasangan kunci ketika Anda membuat sebuah instans.

Versi Chef

Ini menunjukkan versi Chef yang telah Anda pilih.

Untuk informasi selengkapnya tentang versi Chef, lihatVersi Chef.

Gunakan buku masak Chef khusus

Apakah akan menginstal buku masak Chef khusus Anda pada instans tumpukan.

Untuk Chef 12, pengaturan defaultnya adalah Ya. Untuk Chef 11, Pengaturan defaultnya adalah No. Opsi Ya menampilkan beberapa pengaturan tambahan yang menyediakan AWS OpsWorks

Stacks informasi yang dibutuhkan untuk menyebarkan buku masak khusus dari repositori mereka ke instance tumpukan, seperti URL repositori. Detailnya tergantung pada repositori mana yang Anda gunakan untuk buku masak Anda. Untuk informasi selengkapnya, lihat Memasang Buku Masak Kustom.

Warna tumpukan

(Opsional) Rona yang digunakan untuk mewakili tumpukan pada konsol AWS OpsWorks Stacks. Anda dapat menggunakan warna yang berbeda untuk tumpukan yang berbeda untuk membantu membedakan, misalnya, antara pengembangan, pementasan, dan tumpukan produksi.

Tanda tumpukan

Anda dapat menerapkan tag di tingkat tumpukan dan lapisan. Saat Anda membuat tag, Anda menerapkan tag ke setiap sumber daya dalam struktur yang ditandai. Misalnya, jika Anda menerapkan tag ke tumpukan, Anda menerapkan tag ke setiap lapisan, dan di dalam setiap lapisan, ke setiap instance, volume Amazon EBS, atau penyeimbang beban Elastic Load Balancing di layer. Untuk informasi selengkapnya tentang cara mengaktifkan tag Anda dan menggunakannya untuk melacak dan mengelola biaya sumber daya AWS OpsWorks Stacks Anda, lihat Menggunakan Tag Alokasi Biaya dan Mengaktifkan Tag Alokasi Biaya yang Ditentukan Pengguna di Panduan Pengguna Billing and Cost Management. Untuk informasi lebih lanjut tentang penandaan di AWS OpsWorks Tumpukan, lihat. Tanda

Opsi Lanjutan

Untuk pengaturan lanjutan, klik Advanced **>>** untuk menampilkan bagian Advanced options and Security.

Bagian Opsi lanjutan memiliki opsi berikut:

Jenis perangkat root default

Menentukan jenis penyimpanan yang akan digunakan untuk volume root instance. Untuk informasi selengkapnya, lihat Penyimpanan.

- Tumpukan Linux menggunakan volume root Amazon EBS secara default, tetapi Anda juga dapat menentukan volume root yang didukung penyimpanan instans.
- Tumpukan Windows harus menggunakan volume root Amazon yang didukung.

Peran IAM

(Opsional) Peran AWS Identity and Access Management (IAM) tumpukan, yang digunakan AWS OpsWorks Stacks untuk berinteraksi dengan AWS atas nama Anda.

Profil instans IAM default

(Opsional) Peran IAM default yang akan dikaitkan dengan EC2 instans Amazon stack. Peran ini memberikan izin ke aplikasi yang berjalan pada instance stack untuk mengakses sumber daya AWS seperti bucket S3.

- Untuk memberikan izin khusus ke aplikasi, pilih profil instans (peran) yang ada yang memiliki kebijakan yang sesuai.
- Awalnya, peran profil tidak memberikan izin, tetapi Anda dapat menggunakan konsol IAM, API, atau CLI untuk melampirkan kebijakan yang sesuai. Untuk informasi selengkapnya, lihat Menentukan Izin untuk Aplikasi yang Berjalan pada instance EC2 .

Wilayah titik akhir API

Pengaturan ini mengambil nilainya dari wilayah yang Anda pilih dalam pengaturan dasar tumpukan. Anda dapat memilih dari titik akhir regional berikut.

- Wilayah AS Timur (N. Virginia)
- Wilayah AS Timur (Ohio)
- Wilayah AS Barat (Oregon)
- Wilayah Barat AS (California U.)
- Wilayah Kanada (Tengah) (hanya API; tidak tersedia untuk tumpukan yang dibuat di AWS Management Console
- Wilayah Asia Pasifik (Mumbai)
- Wilayah Asia Pasifik (Singapura)
- Wilayah Asia Pasifik (Sydney)
- Wilayah Asia Pacific (Tokyo)
- Wilayah Asia Pasifik (Seoul)
- Wilayah Eropa (Frankfurt)
- Wilayah Eropa (Irlandia)
- Wilayah Eropa (London)
- Wilayah Eropa (Paris)
- Wilayah Amerika Selatan (Sao Paulo)

Tumpukan yang dibuat dalam satu titik akhir API tidak tersedia di titik akhir API lainnya. Karena pengguna AWS OpsWorks Stacks juga spesifik wilayah, jika Anda ingin pengguna AWS OpsWorks Stacks di salah satu wilayah titik akhir ini mengelola tumpukan di wilayah titik akhir lain, Anda harus mengimpor pengguna ke titik akhir yang terkait dengan tumpukan. Untuk informasi selengkapnya tentang mengimpor pengguna, lihat Mengimpor Pengguna ke AWS OpsWorks Tumpukan.

Tema Hostname

(Opsional) String yang digunakan untuk menghasilkan nama host default untuk setiap instance. Nilai defaultnya adalah Layer Dependent, yang menggunakan nama pendek dari layer instance dan menambahkan nomor unik untuk setiap instance. Misalnya, root tema Load Balancer yang bergantung pada peran adalah “lb”. Contoh pertama yang Anda tambahkan ke layer bernama “lb1", yang kedua “lb2", dan seterusnya.

OpsWorks Versi agen

(Opsional) Apakah akan memperbarui agen AWS OpsWorks Stacks secara otomatis saat versi baru tersedia, atau menggunakan versi agen tertentu dan memperbaruinya secara manual. Fitur ini tersedia di tumpukan Chef 11.10 dan Chef 12. Pengaturan default adalah Pembaruan manual, diatur ke versi agen terbaru.

AWS OpsWorks Stacks menginstal agen pada setiap instance yang berkomunikasi dengan layanan dan menangani tugas seperti memulai Chef berjalan sebagai respons terhadap peristiwa siklus hidup. Agen ini diperbarui secara berkala. Anda memiliki dua opsi untuk menentukan versi agen untuk tumpukan Anda.

- Pembaruan otomatis — AWS OpsWorks Tumpukan secara otomatis menginstal setiap versi agen baru pada instans tumpukan segera setelah pembaruan tersedia.
- Pembaruan manual - AWS OpsWorks Tumpukan menginstal versi agen tertentu pada instans tumpukan.

  AWS OpsWorks Tumpukan memposting pesan di halaman tumpukan saat versi agen baru tersedia, tetapi tidak memperbarui instance tumpukan. Untuk memperbarui agen, Anda harus memperbarui pengaturan tumpukan secara manual untuk menentukan versi agen baru dan AWS OpsWorks Stacks kemudian akan memperbarui instance tumpukan.

Anda dapat mengganti setelan Versi OpsWorks Agen default untuk instance tertentu dengan memperbarui konfigurasinya. Dalam hal ini, pengaturan instance diutamakan. Misalnya, setelan defaultnya adalah Pembaruan otomatis tetapi Anda menentukan Pembaruan manual untuk instance tertentu. Ketika AWS OpsWorks Stacks merilis versi agen baru, itu akan secara otomatis

memperbarui semua instance tumpukan kecuali yang disetel ke pembaruan Manual. Untuk menginstal versi agen baru pada instance itu, Anda harus memperbarui konfigurasinya secara manual dan menentukan versi baru.

> **Note**
>
> Konsol menampilkan nomor versi agen yang disingkat. Untuk melihat nomor versi lengkap, hubungi describe-agent-versionsperintah AWS CLI atau metode API atau SDK yang setara. Mereka mengembalikan nomor versi lengkap untuk versi agen yang tersedia.

JSON kustom

(Opsional) Satu atau lebih atribut kustom, diformat sebagai struktur JSON. Atribut ini digabungkan ke dalam konfigurasi tumpukan dan atribut penerapan yang diinstal pada setiap instance dan dapat digunakan oleh resep. Anda dapat menggunakan JSON khusus, misalnya, untuk menyesuaikan pengaturan konfigurasi dengan mengganti atribut bawaan yang menentukan pengaturan default. Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom.

Keamanan memiliki satu opsi, Gunakan grup OpsWorks keamanan, yang memungkinkan Anda menentukan apakah akan mengaitkan grup keamanan bawaan AWS OpsWorks Stacks dengan lapisan tumpukan.

AWS OpsWorks Tumpukan menyediakan seperangkat standar grup keamanan bawaan — satu untuk setiap lapisan — yang terkait dengan lapisan secara default. Gunakan grup OpsWorks keamanan memungkinkan Anda menyediakan grup keamanan kustom Anda sendiri. Untuk informasi selengkapnya, lihat Menggunakan Grup Keamanan.

Gunakan grup OpsWorks keamanan memiliki pengaturan berikut:

- Ya - AWS OpsWorks Tumpukan secara otomatis mengaitkan grup keamanan bawaan yang sesuai dengan setiap lapisan (pengaturan default).

  Anda dapat mengaitkan grup keamanan tambahan dengan lapisan setelah membuatnya tetapi Anda tidak dapat menghapus grup keamanan bawaan.
- Tidak - AWS OpsWorks Tumpukan tidak mengaitkan grup keamanan bawaan dengan lapisan.

  Anda harus membuat grup EC2 keamanan yang sesuai dan mengaitkan grup keamanan dengan setiap lapisan yang Anda buat. Namun, Anda masih dapat secara manual mengaitkan grup

keamanan bawaan dengan lapisan pada pembuatan; grup keamanan kustom diperlukan hanya untuk lapisan yang memerlukan pengaturan kustom.

Perhatikan hal berikut:

- Jika grup OpsWorks keamanan Gunakan diatur ke Ya, Anda tidak dapat membatasi pengaturan akses port grup keamanan default dengan menambahkan grup keamanan yang lebih ketat ke lapisan. Dengan beberapa grup keamanan, Amazon EC2 menggunakan pengaturan yang paling permisif. Selain itu, Anda tidak dapat membuat pengaturan yang lebih ketat dengan memodifikasi konfigurasi grup keamanan bawaan. Saat Anda membuat tumpukan, AWS OpsWorks Stacks menimpa konfigurasi grup keamanan bawaan dengan pengaturan standar, sehingga setiap perubahan yang Anda buat akan hilang saat berikutnya Anda membuat tumpukan. Jika lapisan memerlukan pengaturan grup keamanan yang lebih ketat daripada grup keamanan bawaan, atur Gunakan grup OpsWorks keamanan ke Tidak, buat grup keamanan khusus dengan pengaturan pilihan Anda, dan tetapkan ke lapisan pada pembuatan.
- Jika Anda secara tidak sengaja menghapus grup keamanan AWS OpsWorks Stacks dan ingin membuatnya kembali, itu harus duplikat persis dari aslinya, termasuk kapitalisasi nama grup. Alih-alih membuat ulang grup secara manual, kami sarankan agar AWS OpsWorks Stacks melakukan tugas untuk Anda. Cukup buat tumpukan baru di wilayah AWS yang sama—dan VPC, jika AWS OpsWorks ada—dan Stacks akan secara otomatis membuat ulang semua grup keamanan bawaan, termasuk yang Anda hapus. Anda kemudian dapat menghapus tumpukan jika Anda tidak memiliki penggunaan lebih lanjut untuk itu; grup keamanan akan tetap ada.

Menjalankan Stack di VPC

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat mengontrol akses pengguna ke instans tumpukan dengan membuatnya di virtual private cloud (VPC). Misalnya, Anda mungkin tidak ingin pengguna memiliki akses langsung ke server

aplikasi atau database stack Anda dan sebagai gantinya mengharuskan semua lalu lintas publik disalurkan melalui penyeimbang beban elastis.

Prosedur dasar untuk menjalankan tumpukan di VPC adalah:

1. Buat VPC yang dikonfigurasikan dengan tepat, menggunakan konsol atau API Amazon VPC, atau templat. AWS CloudFormation
2. Tentukan ID VPC saat Anda membuat tumpukan.
3. Luncurkan instance tumpukan di subnet yang sesuai.

Berikut ini secara singkat menjelaskan bagaimana VPCs bekerja di AWS OpsWorks Stacks.

Important

Jika Anda menggunakan fitur VPC Endpoint, perhatikan bahwa setiap instans di tumpukan harus dapat menyelesaikan tindakan berikut dari Amazon Simple Storage Service (Amazon S3):

- Instal agen instance.
- Instal aset, seperti Ruby.
- Unggah log jalankan Chef.
- Ambil perintah tumpukan.

Untuk mengaktifkan tindakan ini, Anda harus memastikan bahwa instance tumpukan memiliki akses ke bucket berikut yang cocok dengan wilayah tumpukan. Jika tidak, tindakan sebelumnya akan gagal.
Untuk Chef 12 Linux dan Chef 12.2 Windows, bucket adalah sebagai berikut.

Bucket Agen	Bucket Aset	Bucket Log	Ember DNA
• opsworks-instance-agent-sa-east-1 • opsworks-instance-agent-ap-selatan-1	• opsworks-instance-assets-us-east-2 • opsworks-instance-assets-us-east-1	• opsworks-us-east-2-log • opsworks-us-east-1-log • opsworks-ap-south-1-log	• opsworks-us-east-2-dna • opsworks-us-east-1-dna • opsworks-ap-south-1-dna

Bucket Agen	Bucket Aset	Bucket Log	Ember DNA
• 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-sentral-1 • opsworks-instance-agent-eu-sentral-1 • opsworks-instance-agent-eu-west-1 • opsworks-instance-agent-eu-west-2 • opsworks-instance-agent-eu-west-3 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-us-east-2	• opsworks-instance-assets-ap-selatan-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-sentral-1 • opsworks-instance-assets-eu-sentral-1 • opsworks-instance-assets-eu-west-1 • opsworks-instance-assets-eu-west-2 • opsworks-instance-assets-eu-west-3 • opsworks-instance-assets-sa-east-1	• opsworks-ap-northeast-1-log • opsworks-ap-northeast-2-log • opsworks-ap-southeast-1-log • opsworks-ap-southeast-2-log • opsworks-ca-central-1-log • opsworks-eu-central-1-log • opsworks-eu-west-1-log • opsworks-eu-west-2-log • opsworks-eu-west-3-log • opsworks-sa-east-1-log • opsworks-us-west-1-log • opsworks-us-west-2-log	• opsworks-ap-northeast-1-dna • opsworks-ap-northeast-2-dna • opsworks-ap-southeast-1-dna • opsworks-ap-southeast-2-dna • opsworks-ca-central-1-dna • opsworks-eu-central-1-dna • opsworks-eu-west-1-dna • opsworks-eu-west-2-dna • opsworks-eu-west-3-dna • opsworks-sa-east-1-dna • opsworks-us-west-1-dna • opsworks-us-west-2-dna

Bucket Agen	Bucket Aset	Bucket Log	Ember DNA
• opsworks-instance-agent-us-west-1 • opsworks-instance-agent-us-west-2	• opsworks-instance-assets-us-west-1 • opsworks-instance-assets-us-west-2		

Untuk Chef 11.10 dan versi sebelumnya untuk Linux, bucket adalah sebagai berikut. Tumpukan Chef 11.4 tidak didukung di endpoint regional di luar Wilayah US East (N. Virginia).

Bucket Agen	Bucket Aset	Bucket Log	Ember DNA
• opsworks-instance-agent-us-east-2 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-ap-selatan-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-sentral-1 • opsworks-instance-agent-eu-sentral-1 • opsworks-instance-agent-eu-west-1	• opsworks-instance-assets-us-east-2 • opsworks-instance-assets-us-east-1 • opsworks-instance-assets-ap-selatan-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-sentral-1 • opsworks-instance-assets-eu-sentral-1 • opsworks-instance-assets-eu-west-1	• prod_stage-log	• prod_tahap-dna

Bucket Agen	Bucket Aset	Bucket Log	Ember DNA
• opsworks-instance-agent-eu-west-2 • opsworks-instance-agent-eu-west-3 • opsworks-instance-agent-us-east-1 • opsworks-instance-agent-us-west-1 • opsworks-instance-agent-us-west-2	• opsworks-instance-assets-eu-west-2 • opsworks-instance-assets-eu-west-3 • opsworks-instance-assets-sa-east-1 • opsworks-instance-assets-us-west-1 • opsworks-instance-assets-us-west-2		

Untuk informasi lebih lanjut, lihat VPC endpoint.

> **Note**
>
> Agar AWS OpsWorks Stacks dapat terhubung ke titik akhir VPC yang Anda aktifkan, Anda juga harus mengonfigurasi perutean untuk NAT atau IP publik, karena agen Stacks masih memerlukan AWS OpsWorks akses ke titik akhir publik.

Topik

- Dasar-Dasar VPC
- Buat VPC untuk Stacks Stack AWS OpsWorks

Dasar-Dasar VPC

Untuk pembahasan mendetail tentang VPCs, lihat Amazon Virtual Private Cloud. Secara singkat, VPC terdiri dari satu atau lebih subnet, yang masing-masing berisi satu atau lebih instance. Setiap subnet memiliki tabel routing terkait yang mengarahkan lalu lintas keluar berdasarkan alamat IP tujuannya.

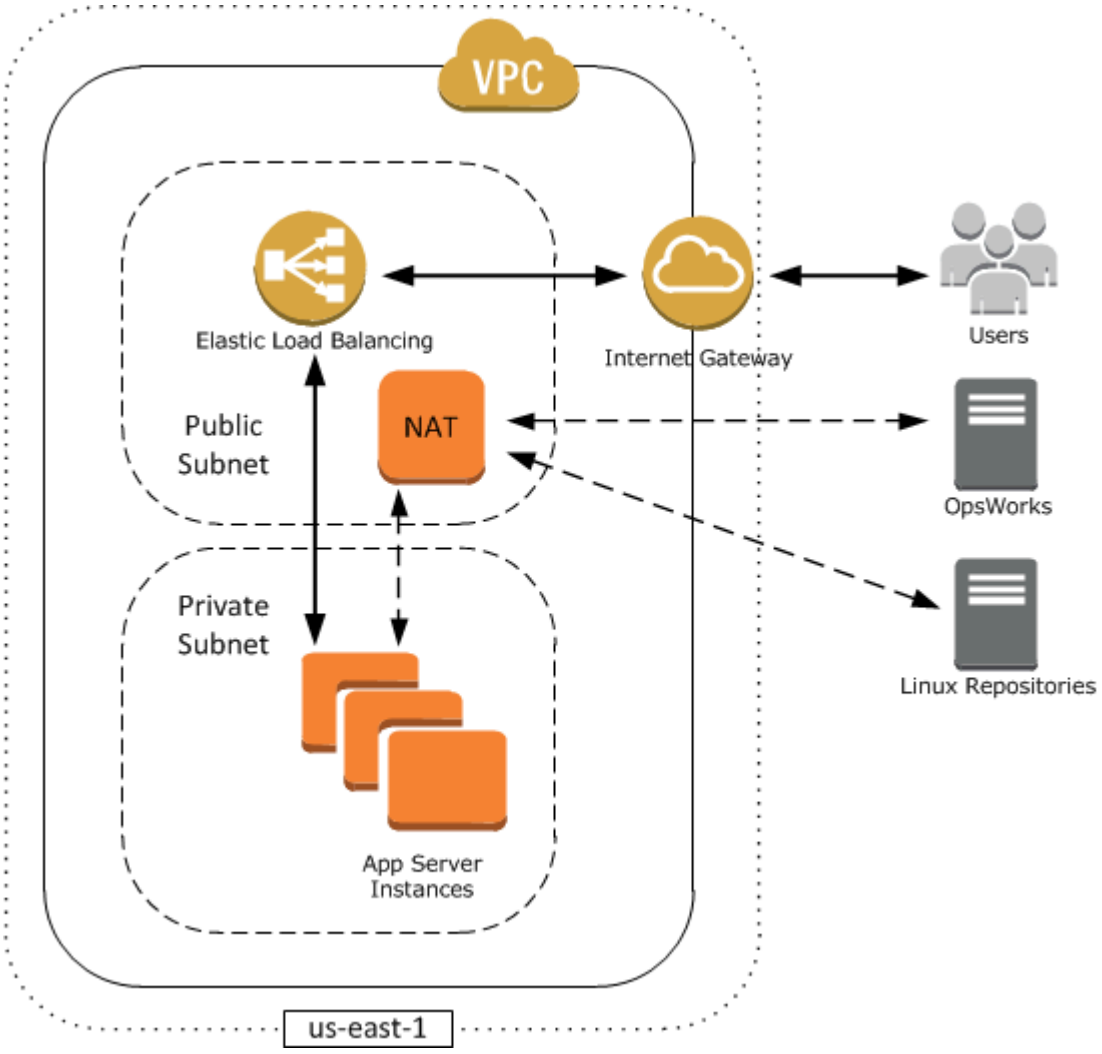
- Instans dalam VPC dapat berkomunikasi satu sama lain secara default, terlepas dari subnetnya. Namun, perubahan pada daftar kontrol akses jaringan (ACLs), kebijakan grup keamanan, atau menggunakan alamat IP statis dapat merusak komunikasi ini.
- Subnet yang instansinya dapat berkomunikasi dengan Internet disebut sebagai subnet publik.
- Subnet yang instansnya hanya dapat berkomunikasi dengan instans lain di VPC dan tidak dapat berkomunikasi langsung dengan Internet disebut sebagai subnet pribadi.

AWS OpsWorks Tumpukan memerlukan VPC untuk dikonfigurasi sehingga setiap instance dalam tumpukan, termasuk instance dalam subnet pribadi, memiliki akses ke titik akhir berikut:

- Salah satu titik akhir layanan AWS OpsWorks Stacks yang tercantum di bagian “Dukungan Wilayah”. Memulai dengan AWS OpsWorks Stacks
- Salah satu titik akhir layanan contoh berikut, digunakan oleh agen AWS OpsWorks Stacks. Agen berjalan pada instance pelanggan terkelola untuk bertukar data dengan layanan.
  - opsworks-instance-service.us-northeast-2.amazonaws.com
  - opsworks-instance-service.us-northeast-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-tenggara 1.amazonaws.com
  - opsworks-instance-service.ap-tenggara 2.amazonaws.com
  - opsworks-instance-service.ca-central-1.amazonaws.com
  - opsworks-instance-service.eu-central-1.amazonaws.com
  - opsworks-instance-service.eu-west-1.amazonaws.com
  - opsworks-instance-service.eu-west-2.amazonaws.com

  - opsworks-instance-service.eu-west-3.amazonaws.com
- Amazon S3
- Repositori paket apa pun yang bergantung pada sistem operasi Anda, seperti repositori Amazon Linux atau Ubuntu Linux.
- Aplikasi dan repositori buku masak khusus Anda.

Ada berbagai cara untuk mengonfigurasi VPC untuk menyediakan konektivitas ini. Berikut ini adalah contoh sederhana bagaimana Anda dapat mengonfigurasi VPC untuk tumpukan server aplikasi AWS OpsWorks Stacks.



VPC ini memiliki beberapa komponen:

Subnet

VPC memiliki dua subnet, satu publik dan satu pribadi.

- Subnet publik berisi penyeimbang beban dan perangkat penerjemahan alamat jaringan (NAT), yang dapat berkomunikasi dengan alamat eksternal dan instans di subnet pribadi.
- Subnet pribadi berisi server aplikasi, yang dapat berkomunikasi dengan NAT dan penyeimbang beban di subnet publik tetapi tidak dapat berkomunikasi langsung dengan alamat eksternal.

Gateway internet

Gateway Internet memungkinkan instance dengan alamat IP publik, seperti penyeimbang beban, untuk berkomunikasi dengan alamat di luar VPC.

Penyeimbang beban

Penyeimbang beban Elastic Load Balancing mengambil lalu lintas masuk dari pengguna, mendistribusikannya ke server aplikasi di subnet pribadi, dan mengembalikan respons kepada pengguna.

NAT

Perangkat (NAT) menyediakan server aplikasi dengan akses Internet terbatas, yang biasanya digunakan untuk tujuan seperti mengunduh pembaruan perangkat lunak dari repositori eksternal. Semua instance AWS OpsWorks Stacks harus dapat berkomunikasi dengan AWS OpsWorks Stacks dan dengan repositori Linux yang sesuai. Salah satu cara untuk menangani masalah ini adalah dengan menempatkan perangkat NAT dengan alamat IP Elastis terkait di subnet publik. Anda kemudian dapat merutekan lalu lintas keluar dari instance di subnet pribadi melalui NAT.

> **Note**
>
> Sebuah instance NAT tunggal menciptakan satu titik kegagalan dalam lalu lintas keluar subnet pribadi Anda. Anda dapat meningkatkan keandalan dengan mengonfigurasi VPC dengan sepasang instance NAT yang mengambil alih satu sama lain jika salah satu gagal. Untuk informasi selengkapnya, lihat Ketersediaan Tinggi untuk Instans NAT Amazon VPC. Anda dapat juga menggunakan gateway NAT. Untuk informasi selengkapnya, lihat NAT di Panduan Pengguna Amazon VPC.

Konfigurasi VPC yang optimal tergantung pada AWS OpsWorks tumpukan Stacks Anda. Berikut ini adalah beberapa contoh kapan Anda mungkin menggunakan konfigurasi VPC tertentu. Untuk contoh skenario VPC lainnya, lihat Skenario untuk Menggunakan Amazon VPC.

Bekerja dengan satu instance di subnet publik

Jika Anda memiliki tumpukan instans tunggal tanpa sumber daya pribadi terkait—seperti instans Amazon RDS yang seharusnya tidak dapat diakses publik — Anda dapat membuat VPC dengan satu subnet publik dan menempatkan instance di subnet tersebut. Jika Anda tidak menggunakan VPC default, Anda harus memiliki lapisan instance yang menetapkan alamat IP Elastis ke instance. Untuk informasi selengkapnya, lihat OpsWorks Dasar-dasar Lapisan.

Bekerja dengan sumber daya pribadi

Jika Anda memiliki sumber daya yang seharusnya tidak dapat diakses publik, Anda dapat membuat VPC dengan satu subnet publik dan satu subnet privat. Misalnya, dalam lingkungan penskalaan otomatis load-balanced, Anda dapat menempatkan semua EC2 instans Amazon di subnet pribadi dan penyeimbang beban di subnet publik. Dengan begitu EC2 instans Amazon tidak dapat diakses langsung dari Internet; semua lalu lintas yang masuk harus diarahkan melalui penyeimbang beban.

Subnet pribadi mengisolasi instance dari akses pengguna EC2 langsung Amazon, tetapi mereka masih harus mengirim permintaan keluar ke AWS dan repositori paket Linux yang sesuai. Untuk mengizinkan permintaan tersebut, Anda dapat, misalnya, menggunakan perangkat terjemahan alamat jaringan (NAT) dengan alamat IP Elastisnya sendiri dan kemudian merutekan lalu lintas keluar instans melalui NAT. Anda dapat menempatkan NAT di subnet publik yang sama dengan penyeimbang beban, seperti yang ditunjukkan pada contoh sebelumnya.

- Jika Anda menggunakan database back-end seperti instans Amazon RDS, Anda dapat menempatkan instance tersebut di subnet pribadi. Untuk instans Amazon RDS, Anda harus menentukan setidaknya dua subnet berbeda di Availability Zone yang berbeda.
- Jika Anda memerlukan akses langsung ke instance di subnet pribadi—misalnya, Anda ingin menggunakan SSH untuk masuk ke instance—Anda dapat menempatkan host bastion di subnet publik yang meminta proxy dari Internet.

Memperluas jaringan Anda sendiri ke AWS

Jika Anda ingin memperluas jaringan Anda sendiri ke cloud dan juga mengakses secara langsung Internet dari VPC Anda, Anda dapat membuat gateway VPN. Untuk informasi selengkapnya, lihat Skenario 3: VPC dengan Subnet Publik dan Akses VPN Perangkat Keras.

Buat VPC untuk Stacks Stack AWS OpsWorks

Bagian ini menunjukkan cara membuat VPC untuk tumpukan AWS OpsWorks Stacks dengan menggunakan contoh template AWS. CloudFormation Anda dapat mengunduh template di

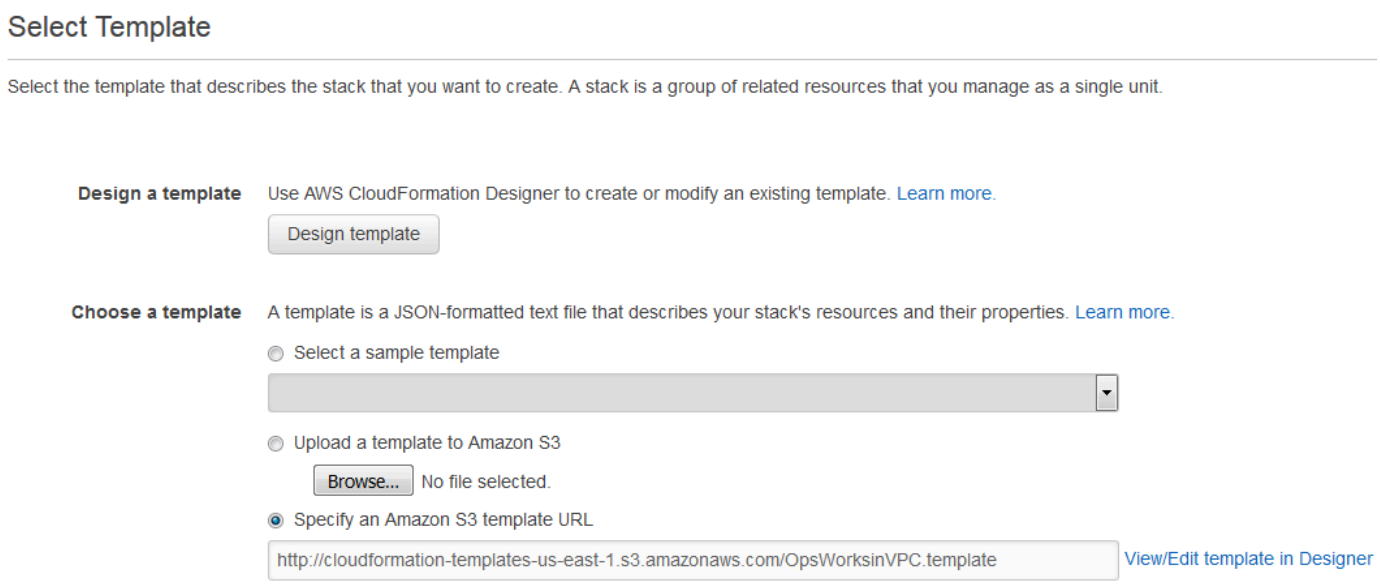
OpsWorksVPCtemplatesfile.zip. Untuk informasi selengkapnya tentang cara membuat VPC secara manual seperti yang dibahas dalam topik ini, lihat Skenario 2: VPC dengan Subnet Publik dan Privat. Untuk detail tentang cara mengkonfigurasi tabel routing, grup keamanan, dan sebagainya, lihat contoh template.

> **Note**
>
> Secara default, AWS OpsWorks Stacks menampilkan nama subnet dengan menggabungkan rentang CIDR dan Availability Zone mereka, seperti. `10.0.0.1/24 - us-east-1b` Untuk membuat nama lebih mudah dibaca, buat tag untuk setiap subnet dengan Key set ke **`Name`** dan Value diatur ke nama subnet. AWS OpsWorks Tumpukan kemudian menambahkan nama subnet ke nama default. Misalnya, subnet pribadi dalam contoh berikut memiliki tag dengan Nama diatur ke**`Private`**, yang OpsWorks menampilkan sebagai`10.0.0.1/24 us-east - 1b - Private`.

Anda dapat meluncurkan template VPC menggunakan AWS CloudFormation konsol hanya dengan beberapa langkah. Prosedur berikut menggunakan templat contoh untuk membuat VPC di US East (N. Virginia). Untuk petunjuk tentang cara menggunakan template untuk membuat VPC di wilayah lain, lihat catatan yang mengikuti prosedur.

Untuk membuat VPC

1. Buka AWS CloudFormation konsol, pilih wilayah US East (Virginia N.), dan pilih Create Stack.
2. Pada halaman Select Template, pilih Upload a template. Jelajahi **`OpsWorksinVPC.template`** file yang Anda unduh di OpsWorksVPCtemplatesfile.zip. Pilih Lanjutkan.



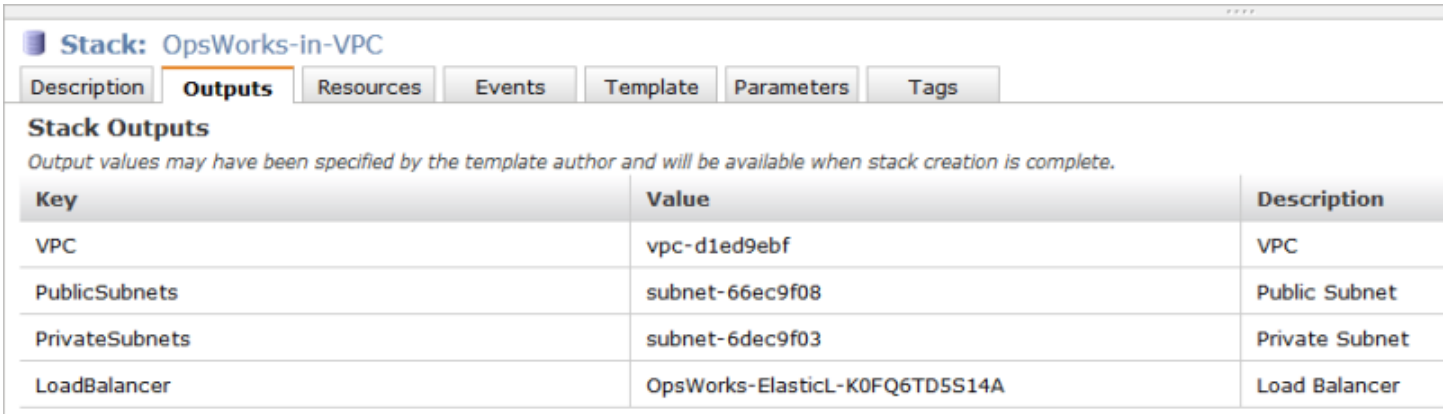
Anda juga dapat meluncurkan tumpukan ini dengan membuka AWS CloudFormation Sample Templates, menemukan template AWS OpsWorks VPC Stacks, dan memilih Launch Stack.

3. Pada halaman Tentukan Parameter, terima nilai default dan pilih Lanjutkan.
4. Pada halaman Tambahkan Tag, buat tag dengan Kunci diatur ke **`Name`** dan Nilai diatur ke nama VPC. Tag ini akan memudahkan untuk mengidentifikasi VPC Anda saat Anda membuat AWS OpsWorks tumpukan Stacks.
5. Pilih Lanjutkan dan kemudian Tutup untuk meluncurkan tumpukan.

Catatan: Anda dapat membuat VPC di wilayah lain dengan menggunakan salah satu pendekatan berikut.

- Buka Menggunakan Template di Wilayah yang Berbeda, pilih wilayah yang sesuai, cari template VPC AWS OpsWorks Stacks, lalu pilih Launch Stack.
- Salin file templat ke sistem Anda, pilih wilayah yang sesuai di AWS CloudFormation konsol, dan gunakan opsi Upload templat ke Amazon S3 wizard Buat Stack untuk mengunggah templat dari sistem Anda.

Contoh template mencakup output yang menyediakan VPC, subnet, dan IDs penyeimbang beban yang Anda perlukan untuk membuat tumpukan Stacks. AWS OpsWorks Anda dapat melihatnya dengan memilih tab Output di bagian bawah jendela AWS CloudFormation konsol.



Perbarui Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda membuat tumpukan, Anda dapat memperbarui konfigurasi kapan saja. Pada halaman Stack, klik Stack Settings, dan kemudian klik Edit, yang menampilkan halaman Pengaturan. Buat perubahan yang Anda inginkan dan klik Simpan.

Pengaturannya sama dengan yang dibahas diMembuat Tumpukan Baru. Lihat topik itu untuk detailnya. Namun, perhatikan hal berikut:

- Anda tidak dapat mengubah wilayah atau ID VPC.
- Jika tumpukan Anda berjalan di VPC, pengaturannya menyertakan pengaturan subnet Default, yang mencantumkan subnet VPC. Jika tumpukan Anda tidak berjalan di VPC, setelan tersebut diberi label Default Availability Zones, dan mencantumkan Availability Zones di wilayah tersebut.
- Anda dapat mengubah sistem operasi default, tetapi Anda tidak dapat menentukan sistem operasi Linux untuk tumpukan Windows, atau Windows untuk tumpukan Linux.
- Jika Anda mengubah salah satu pengaturan instans default, seperti tema Hostname atau kunci SSH Default, nilai baru hanya berlaku untuk setiap instance baru yang Anda buat, bukan untuk instance yang ada.

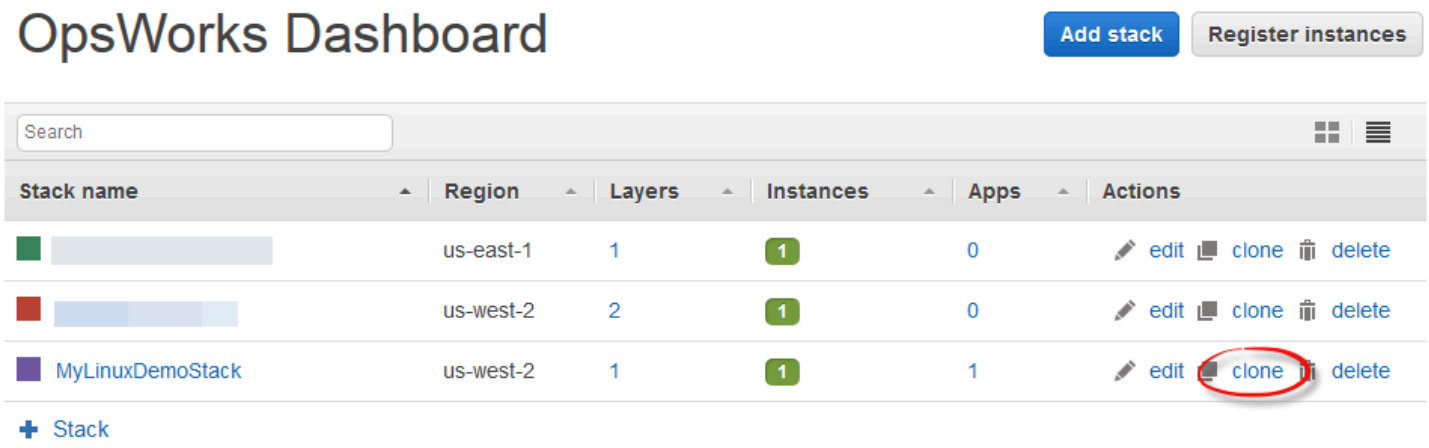
- Mengubah Nama mengubah nama yang ditampilkan oleh konsol; itu tidak mengubah nama pendek yang mendasari yang digunakan AWS OpsWorks Stacks untuk mengidentifikasi tumpukan.
- Sebelum Anda mengubah Gunakan grup OpsWorks keamanan dari Ya ke Tidak, setiap lapisan harus memiliki setidaknya satu grup keamanan selain grup keamanan bawaan lapisan. Untuk informasi selengkapnya, lihat Mengedit Konfigurasi OpsWorks Layer.

  AWS OpsWorks Stacks kemudian menghapus grup keamanan bawaan dari setiap lapisan.
- Jika Anda mengubah Gunakan grup OpsWorks keamanan dari Tidak ke Ya, AWS OpsWorks Stacks menambahkan grup keamanan bawaan yang sesuai ke setiap lapisan tetapi tidak menghapus grup keamanan yang ada.

Klon Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Terkadang berguna untuk membuat beberapa salinan tumpukan. Misalnya, Anda mungkin ingin menambahkan redundansi sebagai pemulihan bencana atau tindakan pencegahan, atau Anda mungkin menggunakan tumpukan yang ada sebagai titik awal untuk tumpukan baru. Pendekatan paling sederhana adalah mengkloning tumpukan sumber. Di dasbor AWS OpsWorks Stacks, di kolom Actions dari baris untuk tumpukan yang ingin Anda kloning, pilih clone, yang membuka halaman tumpukan Clone.



Awalnya, pengaturan untuk tumpukan kloning identik dengan pengaturan untuk tumpukan sumber kecuali bahwa kata “salin” ditambahkan ke nama tumpukan. Untuk informasi tentang pengaturan ini, lihat Membuat Tumpukan Baru. Ada juga dua pengaturan opsional tambahan:

Izin

Jika semua izin dipilih (default), izin tumpukan sumber diterapkan ke tumpukan kloning.

Aplikasi

Daftar aplikasi yang digunakan di tumpukan sumber. Untuk setiap aplikasi yang terdaftar, jika kotak centang yang sesuai dipilih (default), aplikasi akan di-deploy ke tumpukan kloning.

> **Note**
>
> Anda tidak dapat mengkloning tumpukan dari satu titik akhir regional ke titik akhir regional lainnya; misalnya, Anda tidak dapat mengkloning tumpukan dari Wilayah AS Barat (Oregon) (us-west-2) ke Wilayah Asia Pasifik (Mumbai) (ap-south-1).

Ketika Anda telah menyelesaikan pengaturan, pilih Clone stack. AWS OpsWorks Stacks membuat tumpukan baru yang terdiri dari lapisan tumpukan sumber dan opsional aplikasi dan izinnya. Lapisan memiliki konfigurasi yang sama dengan aslinya, tunduk pada modifikasi apa pun yang Anda buat. Namun, kloning tidak membuat instance apa pun. Anda harus menambahkan satu set instance yang sesuai untuk setiap lapisan tumpukan kloning dan kemudian memulainya. Seperti halnya tumpukan apa pun, Anda dapat melakukan tugas manajemen normal pada tumpukan kloning, seperti menambahkan, menghapus, atau memodifikasi lapisan atau menambahkan dan menerapkan aplikasi.

Untuk membuat tumpukan kloning operasional, mulai instance. AWS OpsWorks Stacks mengatur dan mengonfigurasi setiap instance sesuai dengan keanggotaan lapisannya. Ini juga menyebarkan aplikasi apa pun, seperti halnya dengan tumpukan baru.

Jalankan AWS OpsWorks Stacks Stack Commands

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks menyediakan satu set perintah stack, yang dapat Anda gunakan untuk melakukan berbagai operasi pada instance stack. Untuk menjalankan perintah stack, klik Run Command pada halaman Stack. Anda kemudian memilih perintah yang sesuai, menentukan opsi apa pun, dan tekan tombol di kanan bawah, yang akan diberi label dengan nama perintah.

> **Note**
>
> AWS OpsWorks Stacks juga mendukung serangkaian perintah penerapan, yang Anda gunakan untuk mengelola penerapan aplikasi. Untuk informasi selengkapnya, lihat Menerapkan Aplikasi.

Anda dapat menjalankan perintah tumpukan berikut pada tumpukan apa pun.

Perbarui Buku Masak Kustom

Memperbarui buku masak khusus instance dengan versi saat ini dari repositori. Perintah ini tidak menjalankan resep apa pun. Untuk menjalankan resep yang diperbarui, Anda dapat menggunakan perintah`Execute Recipes`,`Setup`, atau `Configure` stack, atau menerapkan ulang aplikasi Anda untuk menjalankan resep Deploy. Untuk informasi selengkapnya tentang buku masak kustom, lihatBuku Masak dan Resep.

Jalankan Resep

Mengeksekusi satu set resep tertentu pada instance. Untuk informasi selengkapnya, lihat Resep yang berjalan secara manual.

Pengaturan

Menjalankan resep Pengaturan instans.

Konfigurasi

Menjalankan resep Konfigurasi instans.

> **Note**
>
> Untuk menggunakan Pengaturan atau Konfigurasi untuk menjalankan resep pada sebuah instance, resep harus ditetapkan ke peristiwa siklus hidup yang sesuai untuk lapisan instance. Untuk informasi selengkapnya, lihat Resep Eksekusi.

Anda dapat menjalankan perintah tumpukan berikut hanya pada tumpukan berbasis Linux.

Instal Dependensi

Menginstal paket instans. Mulai dari Chef 12, perintah ini tidak tersedia.

Perbarui Dependensi

(Linux saja. Mulai dari Chef 12, perintah ini tidak tersedia.) Menginstal pembaruan sistem operasi reguler dan pembaruan paket. Detailnya tergantung pada sistem operasi instans. Untuk informasi selengkapnya, lihat Mengelola Pembaruan Keamanan.

Gunakan perintah Upgrade Operating System untuk meng-upgrade instance ke versi Amazon Linux yang baru.

Upgrade Sistem Operasi

(Hanya Linux) Meningkatkan sistem operasi Amazon Linux instans ke versi terbaru. Untuk informasi selengkapnya, lihat AWS OpsWorks Stacks sistem operasi.

> **Important**
>
> Setelah menjalankan Upgrade Operating System, kami sarankan Anda juga menjalankan Setup. Ini memastikan bahwa layanan dimulai ulang dengan benar.

Perintah tumpukan memiliki opsi berikut, beberapa di antaranya hanya muncul untuk perintah tertentu.

Komentar

(Opsional) Masukkan komentar khusus yang ingin Anda tambahkan.

Resep untuk dieksekusi

(Wajib) Pengaturan ini hanya muncul jika Anda memilih perintah Execute Recipes. Masukkan resep yang akan dieksekusi menggunakan *recipe_name* format standar*cookbook_name*::, dipisahkan dengan koma. Jika Anda menentukan beberapa resep, AWS OpsWorks Stacks mengeksekusinya dalam urutan yang tercantum.

Izinkan reboot

(Opsional) Pengaturan ini hanya muncul jika Anda memilih perintah Upgrade Sistem Operasi. Nilai defaultnya adalah Ya, yang mengarahkan AWS OpsWorks Stacks untuk me-reboot instance setelah menginstal pemutakhiran.

Chef Kustom JSON

(Opsional) Pilih Lanjutan untuk menampilkan opsi ini, yang memungkinkan Anda menentukan atribut JSON khusus untuk dimasukkan ke dalam konfigurasi tumpukan dan atribut penerapan.

Instans

(Opsional) Tentukan instans mana yang akan menjalankan perintah. Semua instans online dipilih secara default. Untuk menjalankan perintah pada subset instance, pilih layer atau instance yang sesuai.

> **Note**
>
> Anda mungkin melihat eksekusi execute_recipes yang tidak Anda jalankan terdaftar di halaman Deployment dan Commands. Ini biasanya hasil dari perubahan izin, seperti memberikan atau menghapus izin SSH untuk pengguna. Saat Anda membuat perubahan

seperti itu, AWS OpsWorks Stacks menggunakan execute_recipes untuk memperbarui izin pada instance.

Menggunakan JSON Kustom

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Beberapa tindakan AWS OpsWorks Stacks memungkinkan Anda menentukan JSON kustom, yang mana AWS OpsWorks Stacks menginstal pada instance dan dapat digunakan oleh resep.

Anda dapat menentukan JSON kustom dalam situasi berikut:

- Saat Anda membuat, memperbarui, atau mengkloning tumpukan.

  AWS OpsWorks Stacks menginstal JSON kustom di semua instans untuk semua peristiwa siklus hidup berikutnya.
- Saat Anda menjalankan perintah deployment atau stack.

  AWS OpsWorks Stacks meneruskan JSON kustom ke instance hanya untuk peristiwa itu.

JSON kustom harus diwakili oleh, dan diformat sebagai, objek JSON yang valid. Misalnya:

```
{
  "att1": "value1",
  "att2": "value2"
  ...
}
```

AWS OpsWorks Tumpukan menyimpan JSON kustom di lokasi berikut:

Pada instans Linux:

- `/var/chef/runs/run-ID/attribs.json`
- `/var/chef/runs/run-ID/nodes/hostname.json`

Pada instans Windows:

- `drive:\chef\runs\run-ID\attribs.json`
- `drive:\chef\runs\run-ID\nodes\hostname.json`

> **Note**
>
> Di Chef 11.10 dan versi sebelumnya untuk Linux, JSON kustom terletak di jalur berikut pada instance Linux, instance Windows tidak tersedia, dan tidak ada file. `attribs.json` Log disimpan dalam folder atau direktori yang sama dengan JSON. Untuk informasi selengkapnya tentang JSON kustom di Chef 11.10 dan versi sebelumnya untuk Linux, lihat Mengganti Atribut dengan Log JSON dan Chef Kustom.
> `/var/lib/aws/opsworks/chef/hostname.json`

Di jalur sebelumnya, *run-ID* adalah ID unik yang ditetapkan AWS OpsWorks Stacks ke setiap Chef yang dijalankan pada sebuah instance, dan *hostname* merupakan nama host instance.

Untuk mengakses JSON kustom dari resep Chef, gunakan node sintaks Chef standar.

Misalnya, Anda ingin menentukan setelan sederhana untuk aplikasi yang ingin Anda terapkan, seperti apakah aplikasi awalnya terlihat dan warna latar depan dan latar belakang awal aplikasi. Misalkan Anda mendefinisikan pengaturan aplikasi ini dengan objek JSON sebagai berikut:

```
{
  "state": "visible",
  "colors": {
    "foreground": "light-blue",
    "background": "dark-gray"
  }
}
```

Untuk mendeklarasikan JSON kustom untuk tumpukan:

1. Pada halaman stack, pilih Stack Settings, lalu pilih Edit.
2. Untuk Custom Chef JSON, ketik objek JSON, lalu pilih Simpan.

> **Note**
>
> Anda dapat mendeklarasikan JSON kustom di tingkat penerapan, lapisan, dan tumpukan. Anda mungkin ingin melakukan ini jika Anda ingin beberapa JSON khusus hanya terlihat oleh penerapan atau lapisan individu. Atau, misalnya, Anda mungkin ingin mengganti sementara JSON khusus yang dideklarasikan di tingkat tumpukan dengan JSON khusus yang dideklarasikan di tingkat lapisan. Jika Anda mendeklarasikan JSON kustom di beberapa tingkatan, JSON khusus yang dideklarasikan pada tingkat penerapan akan mengesampingkan JSON kustom apa pun yang dideklarasikan di tingkat lapisan dan tumpukan. JSON khusus yang dideklarasikan pada tingkat lapisan mengesampingkan JSON kustom apa pun yang dideklarasikan hanya pada tingkat tumpukan.
>
> Untuk menggunakan konsol AWS OpsWorks Stacks untuk menentukan JSON kustom untuk penerapan, pada halaman Deploy App, pilih Advanced. Ketik JSON kustom di kotak Custom Chef JSON, lalu pilih Simpan.
>
> Untuk menggunakan konsol AWS OpsWorks Stacks untuk menentukan JSON kustom untuk lapisan, pada halaman Layers, pilih Pengaturan untuk lapisan yang diinginkan. Ketik JSON kustom di kotak Custom JSON, lalu pilih Simpan.
>
> Untuk informasi selengkapnya, lihat Mengedit Konfigurasi OpsWorks Layer dan Menerapkan Aplikasi.

Saat Anda menjalankan perintah deployment atau stack, resep dapat mengambil nilai-nilai kustom ini dengan menggunakan `node` sintaks Chef standar, yang memetakan langsung ke hierarki di objek JSON kustom. Misalnya, kode resep berikut menulis pesan ke log Chef tentang nilai JSON kustom sebelumnya:

```
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']}' **********")
```

Pendekatan ini dapat berguna untuk meneruskan data ke resep. AWS OpsWorks Stacks menambahkan data tersebut ke instance, dan resep dapat mengambil data dengan menggunakan sintaks Chef node standar.

> **Note**
>
> Custom JSON dibatasi hingga 120 KB. Jika Anda membutuhkan kapasitas lebih, sebaiknya simpan sebagian data di Amazon Simple Storage Service (Amazon S3). Resep kustom Anda kemudian dapat menggunakan AWS CLI atau AWS SDK untuk Rubyuntuk mengunduh data dari bucket Amazon S3 ke instans Anda.

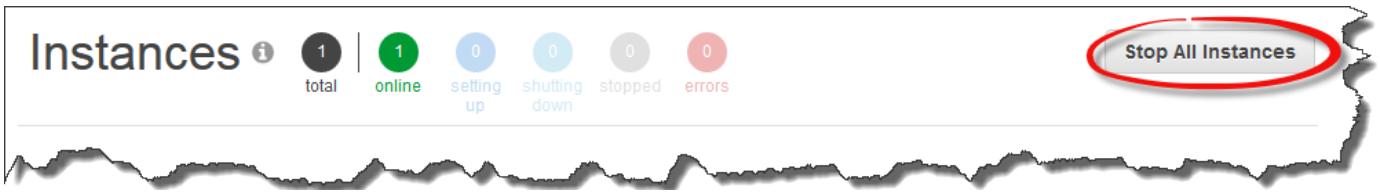
Hapus Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika Anda tidak lagi membutuhkan tumpukan, Anda dapat menghapusnya. Hanya tumpukan kosong yang dapat dihapus; Anda harus terlebih dahulu menghapus semua instance, aplikasi, dan lapisan di tumpukan.

Untuk menghapus tumpukan Anda

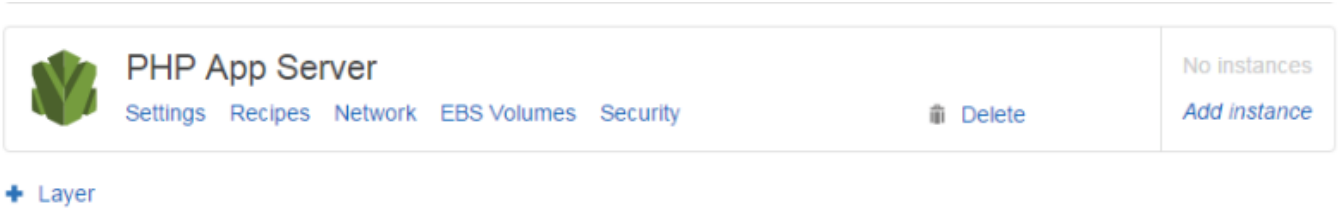
1. Di dasbor AWS OpsWorks Tumpukan, pilih tumpukan yang ingin Anda matikan dan hapus.
2. Di panel navigasi, pilih Instans.
3. Pada halaman Instans, pilih Hentikan semua Instans.



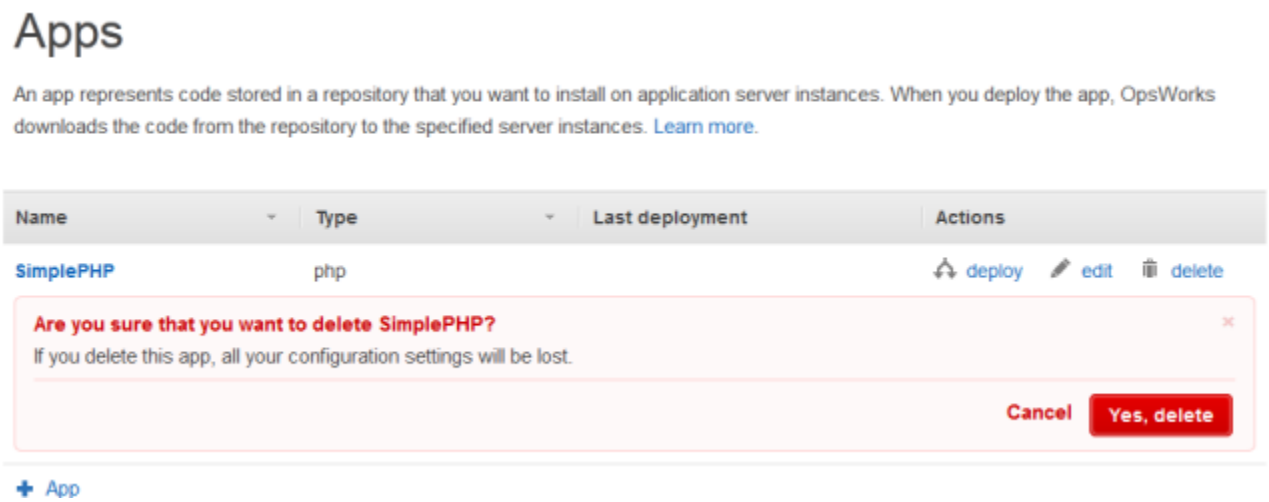
4. Setelah instance berhenti, untuk setiap instance di layer, pilih delete di kolom Actions. Ketika diminta untuk mengonfirmasi, pilih Ya, Hapus.



5. Saat semua instans dihapus, di panel navigasi, pilih Layers.
6. Pada halaman Layers, untuk setiap layer dalam stack, pilih delete. Pada prompt konfirmasi, pilih Ya, Hapus.

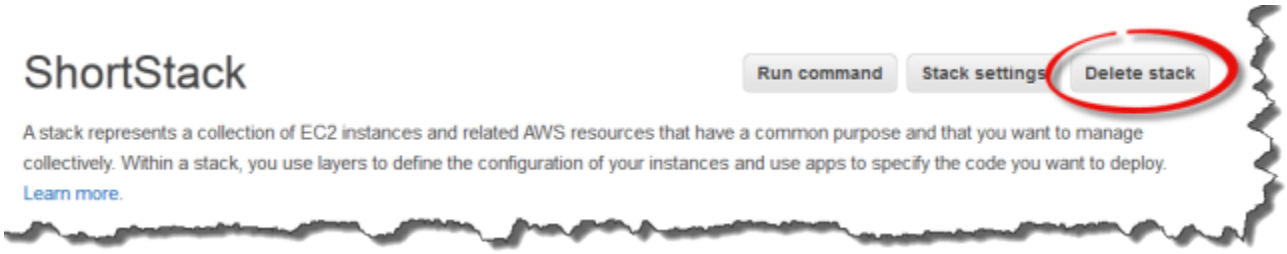


7. Saat semua lapisan dihapus, pada panel navigasi, pilih Apps.
8. Di halaman Aplikasi, untuk setiap aplikasi di tumpukan, pilih hapus di kolom Tindakan. Pada prompt konfirmasi, pilih Ya, Hapus.



9. Saat semua aplikasi dihapus, pada panel navigasi, pilih Tumpukan.

10. Pada halaman tumpukan, pilih Hapus tumpukan. Pada prompt konfirmasi, pilih Ya, Hapus.



Menghapus AWS Sumber Daya Lain yang Digunakan oleh Stack

Anda menggunakan AWS sumber daya lain dengan AWS OpsWorks Stacks untuk membuat dan mengelola tumpukan Anda. Saat Anda menghapus tumpukan, pertimbangkan juga untuk menghapus sumber daya yang bekerja dengan tumpukan, jika tumpukan lain tidak menggunakannya, dan sumber daya di luar AWS OpsWorks Stacks tidak menggunakannya. Berikut ini adalah alasan yang disarankan untuk membersihkan AWS sumber daya eksternal yang Anda gunakan dengan tumpukan.

- AWS Sumber daya eksternal dapat terus menambah biaya pada akun Anda AWS .
- Sumber daya seperti bucket Amazon S3 dapat berisi informasi pengenal pribadi, sensitif, atau rahasia.

> **Important**
>
> Jangan hapus sumber daya ini jika digunakan oleh tumpukan lain. Perhatikan bahwa peran IAM dan grup keamanan bersifat global, jadi tumpukan di wilayah lain mungkin menggunakan sumber daya yang sama ini.

Berikut ini adalah AWS sumber daya lain yang digunakan tumpukan, dan tautan ke informasi tentang cara menghapusnya.

Peran layanan dan profil instance

Saat membuat tumpukan, Anda menentukan peran IAM dan profil instans yang digunakan AWS OpsWorks Stacks untuk membuat sumber daya yang diizinkan atas nama Anda. AWS OpsWorks membuat profil peran dan contoh untuk Anda jika Anda tidak memilih yang sudah ada. Profil peran dan contoh yang AWS OpsWorks dibuat untuk Anda diberi nama `aws-opsworks-`

`service-role` dan`aws-opsworks-ec2-role`, masing-masing. Jika tidak ada tumpukan lain di akun Anda yang menggunakan peran IAM dan profil instans, aman untuk menghapus sumber daya ini. Untuk informasi tentang menghapus peran IAM dan profil instans, lihat Menghapus peran atau profil instans di Panduan Pengguna IAM.

Grup keamanan

Di AWS OpsWorks Stacks, Anda dapat menentukan grup keamanan yang ditentukan pengguna di tingkat lapisan. Anda membuat grup keamanan menggunakan EC2 konsol Amazon atau API. Tumpukan dan lapisan di wilayah lain dapat menggunakan grup keamanan yang sama, karena grup keamanan bersifat global. Anda dapat menghapus grup keamanan jika tidak digunakan oleh AWS sumber daya lain. Untuk informasi selengkapnya tentang cara menghapus grup keamanan, lihat Menghapus grup keamanan di Panduan EC2 Pengguna Amazon.

Volume Amazon EBS

Di AWS OpsWorks Stacks, Anda menambahkan volume EBS di tingkat lapisan, dan mereka dilampirkan ke instance di layer. Anda membuat volume EBS dengan menggunakan konsol EC2 layanan Amazon atau API, lalu melampirkannya ke instance AWS OpsWorks Stacks di tingkat lapisan. Volume EBS khusus untuk zona ketersediaan. Jika Anda tidak lagi menggunakan volume EBS di tumpukan apa pun di wilayah dan zona ketersediaan tertentu, Anda dapat menghapus volume. Untuk informasi selengkapnya tentang cara menghapus volume Amazon EBS, lihat Menghapus Volume Amazon EBS di Panduan Pengguna Amazon EC2 .

bucket Amazon Simple Storage Service (Amazon S3)

Di AWS OpsWorks Stacks, Anda dapat menggunakan bucket Amazon S3 untuk yang berikut ini. Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang cara menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Cara Menghapus Bucket S3? .

- Menyimpan kode aplikasi
- Menyimpan buku masak dan resep
- CloudTrail log, jika Anda telah mengaktifkan CloudTrail login di AWS OpsWorks Stacks
- Amazon CloudWatch Logs stream, jika Anda telah mengaktifkannya di Stacks AWS OpsWorks

Alamat IP elastis

Jika Anda mendaftarkan alamat IP Elastis dengan AWS OpsWorks Stacks, dan Anda tidak lagi memerlukan alamat IP Elastis, Anda dapat merilis alamat IP Elastis.

Penyeimbang beban Elastic Load Balancing

Jika Anda tidak lagi membutuhkan penyeimbang beban klasik Elastic Load Balancing yang telah Anda gunakan dengan lapisan di tumpukan Anda, Anda dapat menghapusnya. Untuk informasi selengkapnya, lihat Menghapus Load Balancer Anda di Panduan Pengguna untuk Classic Load Balancer.

Instans Amazon Relational Database Service (Amazon RDS)

Jika Anda mendaftarkan instans Amazon RDS database (DB) dengan AWS OpsWorks Stacks, dan Anda tidak lagi membutuhkannya, Anda dapat menghapus instans DB. Untuk informasi selengkapnya tentang cara menghapus instans DB, lihat Menghapus Instans DB di Panduan Pengguna Amazon RDS.

Klaster Amazon Elastic Container Service (Amazon ECS)

Jika tumpukan Anda menyertakan lapisan cluster ECS, dan Anda tidak lagi menggunakan cluster ECS yang terdaftar dengan lapisan, Anda dapat menghapus cluster ECS. Untuk informasi selengkapnya tentang cara menghapus klaster ECS, lihat Menghapus Cluster di Panduan Pengembang Amazon ECS.

Lapisan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setiap tumpukan berisi satu atau lebih lapisan, yang masing-masing mewakili komponen tumpukan, seperti penyeimbang beban atau satu set server aplikasi.

Saat Anda bekerja dengan lapisan AWS OpsWorks Tumpukan, ingatlah hal berikut:

- Setiap lapisan dalam tumpukan harus memiliki setidaknya satu instance dan secara opsional dapat memiliki beberapa instance.

- Setiap instance dalam tumpukan harus menjadi anggota dari setidaknya satu lapisan, kecuali untuk instance terdaftar.

  Anda tidak dapat mengonfigurasi instance secara langsung, kecuali untuk beberapa pengaturan dasar seperti kunci SSH dan nama host. Anda harus membuat dan mengonfigurasi lapisan yang sesuai, dan menambahkan instans ke lapisan.

EC2 Instans Amazon secara opsional dapat menjadi anggota beberapa lapisan. Dalam hal ini, AWS OpsWorks Stacks menjalankan resep untuk menginstal dan mengkonfigurasi paket, menyebarkan aplikasi, dan sebagainya untuk setiap lapisan instance.

Dengan menetapkan instance ke beberapa lapisan, Anda dapat, misalnya melakukan hal berikut:

- Kurangi biaya dengan menghosting server database dan penyeimbang beban pada satu instance.
- Gunakan salah satu server aplikasi Anda untuk administrasi.

  Buat lapisan administratif khusus dan tambahkan salah satu instance server aplikasi ke lapisan itu. Resep lapisan administratif mengkonfigurasi instance server aplikasi untuk melakukan tugas administratif, dan menginstal perangkat lunak tambahan yang diperlukan. Contoh server aplikasi lainnya hanyalah server aplikasi.

Bagian ini menjelaskan cara bekerja dengan lapisan.

Topik

- OpsWorks Dasar-dasar Lapisan
- Lapisan Elastic Load Balancing
- Lapisan Layanan Amazon RDS
- Lapisan Klaster ECS
- Lapisan AWS OpsWorks Tumpukan Kustom
- Instalasi Paket Sistem Operasi Per-layer

OpsWorks Dasar-dasar Lapisan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara untuk melakukan operasi yang umum untuk semua lapisan AWS OpsWorks Stacks.

Topik

- Membuat OpsWorks Layer
- Mengedit Konfigurasi OpsWorks Layer
- Menggunakan Penyembuhan Otomatis untuk Mengganti Instans Gagal
- Menghapus Layer OpsWorks

Membuat OpsWorks Layer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Saat Anda membuat tumpukan baru, Anda melihat halaman berikut:



Untuk menambahkan OpsWorks layer pertama

1. Klik Tambahkan Layer.
2. Pada halaman Add Layer, pilih layer yang sesuai, yang menampilkan opsi konfigurasi layer.
3. Konfigurasikan layer dengan tepat dan klik Add Layer untuk menambahkannya ke tumpukan. Bagian berikut menjelaskan cara mengonfigurasi berbagai lapisan.

   > **Note**
   >
   > Halaman Add Layer hanya menampilkan pengaturan konfigurasi yang lebih umum digunakan untuk setiap lapisan. Anda dapat menentukan pengaturan tambahan dengan mengedit lapisan.

4. Tambahkan instans ke layer dan mulai.

   > **Note**
   >
   > Jika sebuah instance adalah anggota dari beberapa lapisan, Anda harus menambahkannya ke semuanya sebelum memulai instance. Anda tidak dapat menambahkan instans online ke lapisan.

Untuk menambahkan lebih banyak layer, buka halaman Layers dan klik + Layer untuk membuka halaman Add Layer.

Ketika Anda memulai sebuah instance, AWS OpsWorks Stacks secara otomatis menjalankan resep Setup dan Deploy untuk setiap layer instance untuk menginstal dan mengkonfigurasi paket yang

sesuai dan menyebarkan aplikasi yang sesuai. Anda dapat menyesuaikan proses penyiapan dan konfigurasi lapisan dalam berbagai cara, seperti dengan menetapkan resep khusus ke peristiwa siklus hidup yang sesuai. AWS OpsWorks Tumpukan menjalankan resep khusus setelah resep standar untuk setiap acara. Untuk informasi selengkapnya, lihat Buku Masak dan Resep.

Bagian khusus lapisan berikut menjelaskan bagaimana menangani Langkah 2 dan 3 untuk berbagai lapisan AWS OpsWorks Stacks. Untuk informasi selengkapnya cara menambahkan instance, lihatMenambahkan Instance ke Layer.

Mengedit Konfigurasi OpsWorks Layer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda membuat layer, beberapa properti (seperti wilayah AWS) tidak dapat diubah, tetapi Anda dapat mengubah sebagian besar konfigurasi lapisan kapan saja. Mengedit layer juga menyediakan akses ke pengaturan konfigurasi yang tidak tersedia pada halaman Add Layer. Pengaturan ini berlaku segera setelah Anda menyimpan konfigurasi baru.

Untuk mengedit OpsWorks lapisan

1. Di panel navigasi, klik Layers.
2. Pada halaman Layers, pilih nama layer untuk membuka halaman detail, yang menunjukkan konfigurasi saat ini.

   > **Note**
   >
   > Memilih salah satu nama di bawah nama lapisan akan membawa Anda langsung ke tab terkait pada halaman detail.

3. Klik Edit dan kemudian pilih tab yang sesuai: Pengaturan Umum, Resep, Jaringan, Volume EBS, atau Keamanan.

Bagian berikut menjelaskan pengaturan pada berbagai tab yang tersedia untuk semua lapisan. Beberapa lapisan memiliki pengaturan khusus lapisan tambahan, yang muncul di bagian atas halaman. Selain itu, beberapa pengaturan hanya tersedia untuk tumpukan berbasis Linux, seperti yang disebutkan.

Topik

- Pengaturan Umum
- Resep
- Jaringan
- Volume EBS
- Keamanan
- CloudWatch Log
- Tanda

Pengaturan Umum

Semua lapisan memiliki pengaturan berikut:

Penyembuhan otomatis diaktifkan

Apakah penyembuhan otomatis diaktifkan untuk instans lapisan. Pengaturan defaultnya adalah Ya.

JSON kustom

Data dalam format JSON yang diteruskan ke resep Chef Anda untuk semua instance di lapisan ini. Anda dapat menggunakan ini, misalnya, untuk meneruskan data ke resep Anda sendiri. Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom.

> **Note**
>
> Anda dapat mendeklarasikan JSON kustom di tingkat penerapan, lapisan, dan tumpukan. Anda mungkin ingin melakukan ini jika Anda ingin beberapa JSON kustom terlihat di seluruh tumpukan atau hanya untuk penerapan individu. Atau, misalnya, Anda mungkin ingin mengganti sementara JSON kustom yang dideklarasikan di tingkat lapisan dengan JSON khusus yang dideklarasikan pada tingkat penerapan. Jika Anda mendeklarasikan JSON kustom di beberapa tingkatan, JSON khusus yang dideklarasikan pada tingkat penerapan akan mengesampingkan JSON kustom apa pun yang dideklarasikan di

tingkat lapisan dan tumpukan. JSON khusus yang dideklarasikan pada tingkat lapisan mengesampingkan JSON kustom apa pun yang dideklarasikan hanya pada tingkat tumpukan.
Untuk menggunakan konsol AWS OpsWorks Stacks untuk menentukan JSON kustom untuk penerapan, pada halaman Deploy App, pilih Advanced. Ketik JSON kustom di kotak Custom Chef JSON, lalu pilih Simpan.
Untuk menggunakan konsol AWS OpsWorks Stacks untuk menentukan JSON kustom untuk tumpukan, pada halaman pengaturan tumpukan, ketik JSON kustom di kotak JSON Kustom, lalu pilih Simpan.
Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom dan Menerapkan Aplikasi.

Waktu shutdown instans

Menentukan berapa lama (dalam detik) AWS OpsWorks Stacks menunggu setelah memicu peristiwa siklus hidup Shutdown sebelum menghentikan atau menghentikan instans Amazon. EC2 Pengaturan default adalah 120 detik. Tujuan dari pengaturan ini adalah untuk memberikan resep Shutdown instance yang cukup waktu untuk menyelesaikan tugas mereka sebelum menghentikan instance. Jika resep Shutdown kustom Anda mungkin memerlukan lebih banyak waktu, ubah pengaturan yang sesuai. Untuk informasi selengkapnya tentang shutdown instans, lihatMenghentikan sebuah Instance.

Pengaturan yang tersisa pada tab ini bervariasi dengan jenis layer dan identik dengan pengaturan pada halaman Add Layer layer.

Resep

Tab Resep mencakup pengaturan berikut.

Resep Chef Kustom

Anda dapat menetapkan resep Chef khusus ke acara siklus hidup lapisan. Untuk informasi selengkapnya, lihat Resep Eksekusi.

Jaringan

Tab Jaringan mencakup pengaturan berikut.

Elastic Load Balancing

Anda dapat memasang penyeimbang beban Elastic Load Balancing ke lapisan apa pun. AWS OpsWorks Stacks kemudian secara otomatis mendaftarkan instance online layer dengan load balancer dan membatalkan pendaftarannya saat offline. Jika Anda telah mengaktifkan fitur pengurasan koneksi penyeimbang beban, Anda dapat menentukan apakah AWS OpsWorks Stacks mendukungnya. Untuk informasi selengkapnya, lihat Lapisan Elastic Load Balancing.

Secara Otomatis Menetapkan Alamat IP

Anda dapat mengontrol apakah AWS OpsWorks Stacks secara otomatis menetapkan alamat IP publik atau Elastis ke instance layer. Inilah yang terjadi saat Anda mengaktifkan opsi ini:

- Misalnya instance yang didukung toko, AWS OpsWorks Stacks secara otomatis menetapkan alamat setiap kali instance dimulai.
- Untuk instans yang didukung Amazon EBS, AWS OpsWorks Stacks secara otomatis menetapkan alamat saat instans dimulai untuk pertama kalinya.
- Jika sebuah instance milik lebih dari satu lapisan, AWS OpsWorks Stacks secara otomatis memberikan alamat jika Anda telah mengaktifkan penugasan otomatis untuk setidaknya satu lapisan,

> **Note**
>
> Jika Anda mengaktifkan penetapan otomatis alamat IP publik, itu hanya berlaku untuk instance baru. AWS OpsWorks Tumpukan tidak dapat memperbarui alamat IP publik untuk instance yang ada.

Jika tumpukan Anda berjalan di VPC, Anda memiliki pengaturan terpisah untuk alamat IP publik dan Elastis. Tabel berikut menjelaskan bagaimana interaksi ini:

		Public IP addresses: Yes	Public IP addresses: No
Elastic IP addresses	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**
>
> Instans harus memiliki cara untuk berkomunikasi dengan layanan AWS OpsWorks Stacks, repositori paket Linux, dan repositori buku masak. Jika Anda tidak menentukan alamat IP publik atau Elastis, VPC Anda harus menyertakan komponen seperti NAT yang memungkinkan instance lapisan untuk berkomunikasi dengan situs eksternal. Untuk informasi selengkapnya, lihat Menjalankan Stack di VPC.

Jika tumpukan Anda tidak berjalan di VPC, alamat IP Elastis adalah satu-satunya pengaturan Anda:

- Ya: Instans menerima alamat IP Elastis saat dimulai untuk pertama kalinya, atau alamat IP publik jika alamat IP Elastis tidak dapat ditetapkan.
- Tidak: Instans menerima alamat IP publik setiap kali dimulai.

Volume EBS

Tab Volume EBS mencakup pengaturan berikut.

Instans yang dioptimalkan EBS

Apakah instans lapisan harus dioptimalkan untuk Amazon Elastic Block Store (Amazon EBS). Untuk informasi selengkapnya, lihat Instans Amazon EBS yang dioptimalkan.

Volume EBS Tambahan

(Hanya Linux) Anda dapat menambahkan volume Amazon EBS ke atau menghapusnya dari instance layer. Ketika Anda memulai sebuah instance, AWS OpsWorks Stacks secara otomatis membuat volume dan melampirkannya ke instance. Anda dapat menggunakan halaman Resources untuk mengelola volume EBS stack. Untuk informasi selengkapnya, lihat Manajemen Sumber Daya.

- Titik pemasangan - (Wajib) Tentukan titik pemasangan atau direktori tempat volume EBS akan dipasang.
- # Disks — (Opsional) Jika Anda menentukan array RAID, jumlah disk dalam array.

  Setiap level RAID memiliki jumlah disk default, tetapi Anda dapat memilih nomor yang lebih besar dari daftar.
- Ukuran total (GiB) - (Diperlukan) Ukuran volume, dalam GiB.

Untuk array RAID, pengaturan ini menentukan ukuran array total, bukan ukuran masing-masing disk.

Tabel berikut menunjukkan ukuran volume minimum dan maksimum yang diizinkan untuk setiap tipe volume.

Jenis volume	Ukuran Minimum (GiB)	Ukuran Maksimum (GiB)
Magnetik	1	1024
IOPS Terprovisi (SSD)	4	16384
Tujuan Umum (SSD)	1	16384
Dioptimalkan (HDD)	500	16384
Cold HDD	500	16384

- Jenis Volume - (Opsional) Tentukan apakah akan membuat SSD Serba Guna, SSD Serba Guna, HDD Keluaran Dioptimalkan, atau volume PIOPS.

  Nilai default-nya adalah Magnetic.
- Terenkripsi - (Opsional) Tentukan apakah akan mengenkripsi isi volume EBS.
- IOPS per disk - (Diperlukan untuk SSD IOPS yang Disediakan dan volume SSD Tujuan Umum) Jika Anda menentukan volume IOPS SSD atau SSD Tujuan Umum, Anda juga harus menentukan IOPS per disk.

  Untuk volume IOPS yang disediakan, Anda dapat menentukan tingkat IOPS saat membuat volume. Rasio IOPS yang disediakan dan ukuran volume yang diminta bisa maksimal 30 (dengan kata lain, volume dengan 3000 IOPS harus minimal 100 GB). Jenis volume General Purpose (SSD) memiliki IOPS dasar ukuran volume x 3 dengan maksimum 10000 IOPS dan dapat meledak hingga 3000 IOPS selama 30 menit.

Saat Anda menambahkan volume ke atau menghapusnya dari lapisan, perhatikan hal berikut:

- Jika Anda menambahkan volume, setiap instans baru mendapatkan volume baru, tetapi AWS OpsWorks Stacks tidak memperbarui instance yang ada.

- Jika Anda menghapus volume, itu hanya berlaku untuk instance baru; instans yang ada mempertahankan volumenya.

Menentukan Titik Gunung

Anda dapat menentukan titik pemasangan apa pun yang Anda inginkan. Namun, perlu diketahui bahwa beberapa titik pemasangan dicadangkan untuk digunakan oleh AWS OpsWorks Stacks atau Amazon EC2 dan tidak boleh digunakan untuk volume Amazon EBS. Jangan gunakan folder sistem Linux yang khas seperti `/home` atau`/etc`.

Titik pemasangan berikut dicadangkan untuk digunakan oleh AWS OpsWorks Stacks.

- `/srv/www`
- `/var/log/apache2`(Ubuntu)
- `/var/log/httpd`(Amazon Linux)
- `/var/log/mysql`
- `/var/www`

Saat instance melakukan booting atau reboot, autofs (daemon automounting) menggunakan titik pemasangan perangkat sementara seperti untuk bind mount. `/media/ephemeral0` Operasi ini dilakukan sebelum volume Amazon EBS dipasang. Untuk memastikan bahwa titik pemasangan volume Amazon EBS Anda tidak bertentangan dengan autofs, jangan tentukan titik pemasangan perangkat sementara. Titik pemasangan perangkat fana yang mungkin bergantung pada jenis instans tertentu, dan apakah itu didukung penyimpanan instans atau didukung Amazon EBS. Untuk menghindari konflik dengan autofs, lakukan hal berikut:

- Verifikasi titik pemasangan perangkat sementara untuk jenis instance tertentu dan backing store yang ingin Anda gunakan.
- Ketahuilah bahwa titik pemasangan yang berfungsi untuk instans yang didukung penyimpanan instans mungkin bertentangan dengan autofs jika Anda beralih ke instans yang didukung Amazon EBS, atau sebaliknya.

> **Note**
>
> Jika ingin mengubah pemetaan perangkat blok penyimpanan instans, Anda dapat membuat AMI kustom. Untuk informasi lebih lanjut, lihat Amazon EC2 Instans Store. Untuk informasi

selengkapnya tentang cara membuat AMI khusus untuk AWS OpsWorks Tumpukan, lihatMenggunakan Custom AMIs.

Berikut ini adalah contoh cara menggunakan resep khusus untuk memastikan bahwa titik pemasangan volume tidak bertentangan dengan autofs. Anda dapat menyesuaikannya sesuai kebutuhan untuk kasus penggunaan khusus Anda.

Untuk menghindari titik pemasangan yang saling bertentangan

1. Tetapkan volume Amazon EBS ke lapisan yang diinginkan tetapi gunakan titik pemasangan seperti `/mnt/workspace` yang tidak akan pernah bertentangan dengan autofs.
2. Terapkan resep kustom berikut, yang membuat direktori aplikasi pada volume Amazon EBS dan menautkannya dari`/srv/www/`. Untuk informasi selengkapnya tentang cara menerapkan resep khusus, lihat Buku Masak dan Resep danMenyesuaikan Tumpukan 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. Tambahkan `depends 'deploy'` baris ke `metadata.rb` file buku masak kustom.
4. Tetapkan resep ini ke acara Setup layer.

Keamanan

Tab Keamanan mencakup pengaturan berikut.

Grup Keamanan

Lapisan harus memiliki setidaknya satu grup keamanan terkait. Anda menentukan cara mengaitkan grup keamanan saat membuat atau memperbarui tumpukan. AWS OpsWorks Tumpukan menyediakan seperangkat standar grup keamanan bawaan.

- Opsi default adalah agar AWS OpsWorks Tumpukan secara otomatis mengaitkan grup keamanan bawaan yang sesuai dengan setiap lapisan.
- Anda juga dapat memilih untuk tidak secara otomatis mengaitkan grup keamanan bawaan dan sebagai gantinya mengaitkan grup keamanan khusus dengan setiap lapisan saat Anda membuat lapisan.

Untuk informasi selengkapnya mengenai Grup Keamanan, lihat Menggunakan Grup Keamanan.

Setelah lapisan dibuat, Anda dapat menggunakan Grup Keamanan untuk menambahkan lebih banyak grup keamanan ke lapisan dengan memilihnya dari daftar grup keamanan khusus. Setelah Anda menambahkan grup keamanan ke layer, AWS OpsWorks Stacks menambahkannya ke semua instance baru. (Perhatikan bahwa instance instance-store yang dimulai ulang akan dimunculkan sebagai instance baru, sehingga mereka juga akan memiliki grup keamanan baru.) AWS OpsWorks Stacks tidak menambahkan grup keamanan ke instance online.

Anda dapat menghapus grup keamanan yang ada dengan mengklik x, sebagai berikut:

- Jika Anda memilih agar AWS OpsWorks Stacks secara otomatis mengaitkan grup keamanan bawaan, Anda dapat menghapus grup keamanan khusus yang ditambahkan sebelumnya dengan mengklik x, tetapi Anda tidak dapat menghapus grup bawaan.
- Jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan bawaan, Anda dapat menghapus grup keamanan yang ada, termasuk yang asli, selama lapisan mempertahankan setidaknya satu grup.

Setelah Anda menghapus grup keamanan dari lapisan, AWS OpsWorks Stacks tidak menambahkannya ke instance baru atau yang dimulai ulang. AWS OpsWorks Stacks tidak menghapus grup keamanan dari instance online.

> **Note**
>
> Jika tumpukan Anda berjalan di VPC, Anda dapat menambahkan atau menghapus grup keamanan untuk instans online dengan menggunakan EC2 konsol Amazon, API, atau CLI. Namun, grup keamanan ini tidak akan terlihat di konsol AWS OpsWorks Stacks. Jika

Anda ingin menghapus grup keamanan, Anda juga harus menggunakan Amazon EC2. Untuk informasi selengkapnya, lihat Grup Keamanan.

Perhatikan hal berikut:

- Anda tidak dapat membatasi pengaturan akses port grup keamanan bawaan dengan menambahkan grup keamanan yang lebih ketat. Ketika ada beberapa grup keamanan, Amazon EC2 menggunakan pengaturan yang paling permisif.
- Anda tidak boleh memodifikasi konfigurasi grup keamanan bawaan. Saat Anda membuat tumpukan, AWS OpsWorks Stacks menimpa konfigurasi grup keamanan bawaan, sehingga setiap perubahan yang Anda buat akan hilang saat berikutnya Anda membuat tumpukan.

Jika Anda menemukan bahwa Anda memerlukan pengaturan grup keamanan yang lebih ketat untuk satu atau beberapa lapisan, lakukan langkah-langkah berikut:

1. Buat grup keamanan khusus dengan pengaturan yang sesuai dan tambahkan ke lapisan yang sesuai.

   Setiap lapisan dalam tumpukan Anda harus memiliki setidaknya satu grup keamanan selain grup bawaan, bahkan jika hanya satu lapisan yang memerlukan pengaturan khusus.
2. Edit konfigurasi tumpukan dan alihkan pengaturan Gunakan grup OpsWorks keamanan ke No.

   AWS OpsWorks Stacks secara otomatis menghapus grup keamanan bawaan dari setiap lapisan.

Untuk informasi selengkapnya tentang grup keamanan, lihat Grup EC2 Keamanan Amazon.

EC2 Profil Instance

Anda dapat mengubah EC2 profil untuk instance layer. Untuk informasi selengkapnya, lihat Menentukan Izin untuk Aplikasi yang Berjalan pada instance EC2 .

CloudWatch Log

Tab CloudWatch Log memungkinkan Anda mengaktifkan atau menonaktifkan CloudWatch Log Amazon. CloudWatch Integrasi log berfungsi dengan tumpukan berbasis Linux Chef 11.10 dan Chef 12. Untuk informasi selengkapnya tentang mengaktifkan integrasi CloudWatch Log dan menentukan log yang ingin Anda kelola di konsol CloudWatch Log, lihat. Menggunakan CloudWatch Log Amazon dengan AWS OpsWorks Tumpukan

Tanda

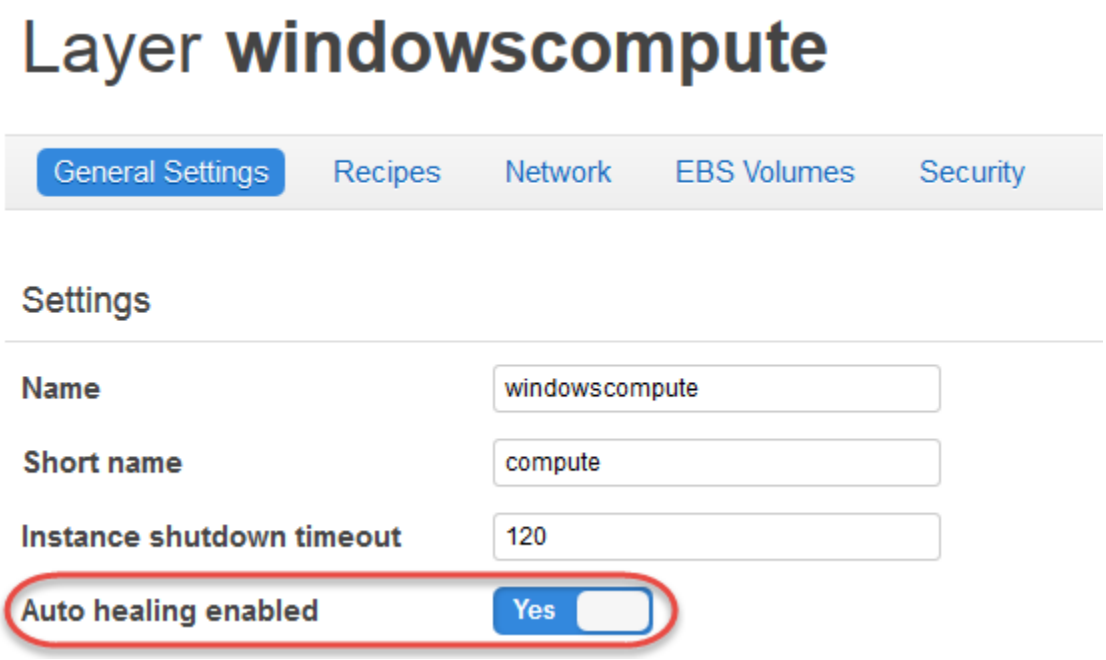
Tab Tag memungkinkan Anda menerapkan tag alokasi biaya ke lapisan Anda. Setelah Anda menambahkan tag, Anda dapat mengaktifkannya di AWS Manajemen Penagihan dan Biaya konsol. Saat Anda membuat tag, Anda menerapkan tag ke setiap sumber daya dalam struktur yang ditandai. Misalnya, jika Anda menerapkan tag ke layer, Anda menerapkan tag ke setiap instance, volume Amazon EBS, atau penyeimbang beban Elastic Load Balancing di layer. Untuk informasi selengkapnya tentang cara mengaktifkan tag Anda dan menggunakannya untuk melacak dan mengelola biaya sumber daya AWS OpsWorks Stacks Anda, lihat Menggunakan Tag Alokasi Biaya dan Mengaktifkan Tag Alokasi Biaya yang Ditentukan Pengguna di Panduan Pengguna Billing and Cost Management. Untuk informasi lebih lanjut tentang penandaan di AWS OpsWorks Tumpukan, lihat. Tanda

Menggunakan Penyembuhan Otomatis untuk Mengganti Instans Gagal

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setiap contoh memiliki agen AWS OpsWorks Stacks yang berkomunikasi secara teratur dengan layanan. AWS OpsWorks Stacks menggunakan komunikasi itu untuk memantau kesehatan contoh. Jika agen tidak berkomunikasi dengan layanan selama lebih dari lima menit, AWS OpsWorks Stacks menganggap instance tersebut gagal.

Penyembuhan otomatis diatur pada tingkat lapisan; Anda dapat mengubah pengaturan penyembuhan otomatis dengan mengedit pengaturan lapisan, seperti yang ditunjukkan pada tangkapan layar berikut.



> **Note**
>
> Instans dapat menjadi anggota beberapa lapisan. Jika salah satu lapisan tersebut menonaktifkan penyembuhan otomatis, AWS OpsWorks Stacks tidak menyembuhkan instance jika gagal.

Jika sebuah layer mengaktifkan auto healing — pengaturan default—AWS OpsWorks Stacks secara otomatis menggantikan instance gagal layer sebagai berikut:

Instans yang didukung toko instans

1. Menghentikan EC2 instans Amazon, dan memverifikasi bahwa instans telah dimatikan.
2. Menghapus data pada volume root.
3. Membuat EC2 instance Amazon baru dengan nama host, konfigurasi, dan keanggotaan layer yang sama.
4. Melampirkan kembali volume Amazon EBS apa pun, termasuk volume yang dilampirkan setelah instans lama awalnya dimulai.
5. Menetapkan Alamat IP privat dan publik baru.
6. Jika instance lama dikaitkan dengan alamat IP Elastis, kaitkan instance baru dengan alamat IP yang sama.

Instans yang didukung Amazon EBS

1. Menghentikan EC2 instans Amazon, dan memverifikasi bahwa itu telah berhenti.
2. Memulai EC2 instance.

Setelah instance yang disembuhkan otomatis kembali online, AWS OpsWorks Stacks memicu peristiwa siklus hidup Konfigurasi pada semua instance tumpukan. Atribut dan konfigurasi tumpukan terkait mencakup alamat IP publik dan alamat IP privat instans. Resep Konfigurasi Kustom dapat memperoleh alamat IP baru dari objek node.

Jika Anda menentukan volume Amazon EBS untuk instance layer, AWS OpsWorks Stacks akan membuat volume baru dan menempelkannya ke setiap instance saat instance dimulai. Jika nanti Anda ingin melepaskan volume dari sebuah instance, gunakan halaman Resources.

Ketika AWS OpsWorks Stacks auto menyembuhkan salah satu instance layer, Stacks menangani volume dengan cara berikut:

- Jika volume dilampirkan ke instance saat instance gagal, volume dan datanya disimpan, dan AWS OpsWorks Stacks menempelkannya ke instance baru.
- Jika volume tidak dilampirkan ke instance saat instance gagal, AWS OpsWorks Stacks membuat volume kosong baru dengan konfigurasi yang ditentukan oleh layer, dan menempelkan volume tersebut ke instance baru.

Penyembuhan otomatis diaktifkan secara default untuk semua lapisan, tetapi Anda dapat mengedit Pengaturan Umum lapisan untuk menonaktifkannya.

Important

Jika Anda mengaktifkan penyembuhan otomatis, pastikan untuk melakukan hal berikut:

- Gunakan hanya konsol AWS OpsWorks Stacks, CLI, atau API untuk menghentikan instance.

  Jika Anda menghentikan instance dengan cara lain, seperti menggunakan EC2 konsol Amazon, AWS OpsWorks Stacks memperlakukan instance sebagai gagal, dan secara otomatis menyembuh instans.
- Gunakan volume Amazon EBS untuk menyimpan data apa pun yang tidak ingin hilang jika instans disembuhkan secara otomatis.

Penyembuhan otomatis menghentikan EC2 instans Amazon lama, yang menghancurkan data apa pun yang tidak disimpan pada volume Amazon EBS. Volume Amazon EBS disambungkan kembali ke instans baru, yang menyimpan data apa pun yang tersimpan.

Menghapus Layer OpsWorks

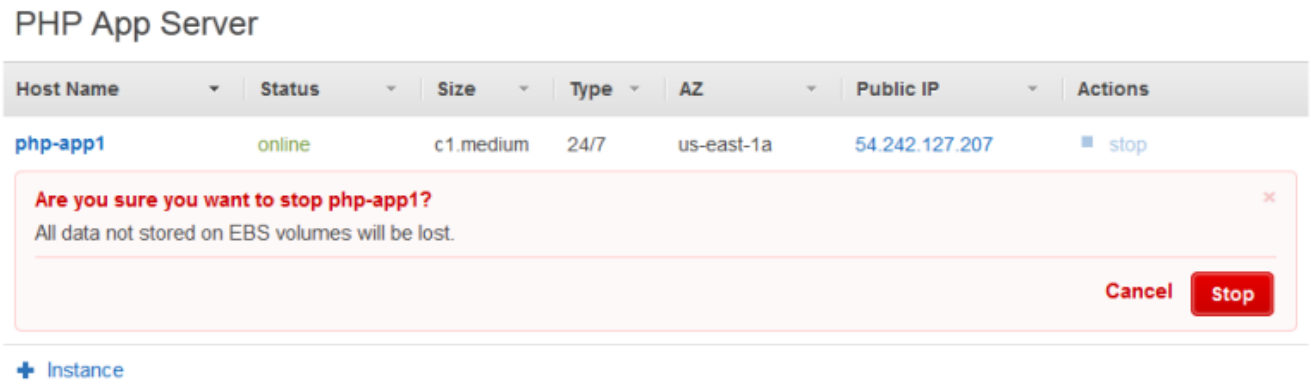
Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

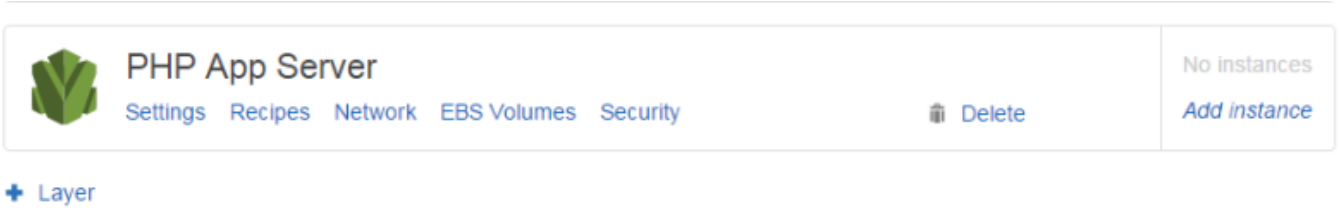
Jika Anda tidak lagi membutuhkan lapisan AWS OpsWorks Tumpukan, Anda dapat menghapusnya dari tumpukan Anda.

Untuk menghapus OpsWorks lapisan

1. Di panel navigasi, klik Instans.
2. Pada halaman Instances, di bawah nama layer yang ingin Anda hapus, klik stop di kolom Actions untuk setiap instance.



3. Setelah setiap instance berhenti, klik hapus untuk menghapusnya dari layer.
4. Di panel navigasi, klik Layers.
5. Pada halaman Layers, pilih Delete.



Lapisan Elastic Load Balancing

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Elastic Load Balancing bekerja agak berbeda dari lapisan AWS OpsWorks Stacks. Alih-alih membuat layer dan menambahkan instance ke dalamnya, Anda menggunakan konsol Elastic Load Balancing atau API untuk membuat penyeimbang beban dan kemudian melampirkannya ke layer yang ada. Selain mendistribusikan lalu lintas ke instance layer, Elastic Load Balancing melakukan hal berikut:

- Mendeteksi EC2 instans Amazon yang tidak sehat dan mengubah rute kemudian lintas ke status sehat yang tersisa hingga instans yang tidak sehat dipulihkan.
- Secara otomatis menskalakan kapasitas penanganan permintaan sebagai respons terhadap lalu lintas yang masuk.
- Jika Anda mengaktifkan pengurasan koneksi, penyeimbang beban berhenti mengirim permintaan baru ke instans yang tidak sehat atau akan dideregistrasi tetapi membuat koneksi tetap hidup, hingga nilai batas waktu yang ditentukan, untuk memungkinkan instance menyelesaikan permintaan dalam penerbangan apa pun.

Setelah Anda memasang penyeimbang beban ke lapisan, AWS OpsWorks Stacks melakukan hal berikut:

- Menderegister semua instance yang terdaftar saat ini.

- Secara otomatis mendaftarkan instance layer saat mereka online dan deregister instance saat offline, termasuk instance berbasis beban dan waktu.
- Secara otomatis memulai perutean permintaan ke instance terdaftar di Availability Zones mereka.

Jika Anda telah mengaktifkan fitur pengurasan koneksi penyeimbang beban, Anda dapat menentukan apakah AWS OpsWorks Stacks mendukungnya. Jika Anda mengaktifkan dukungan pengurasan koneksi (pengaturan default), setelah instance dimatikan, AWS OpsWorks Stacks melakukan hal berikut:

- Memindahkan instans dari penyeimbang beban.

  Penyeimbang beban berhenti mengirim permintaan baru dan mulai menguras koneksi.
- Penundaan memicu peristiwa siklus hidup Shutdown hingga penyeimbang beban menyelesaikan pengurasan koneksi.

Jika Anda tidak mengaktifkan dukungan penguras koneksi, AWS OpsWorks Stacks akan memicu peristiwa Shutdown segera setelah instance dimatikan, meskipun instance masih terhubung ke penyeimbang beban.

Untuk menggunakan Elastic Load Balancing dengan tumpukan, Anda harus terlebih dahulu membuat satu atau beberapa penyeimbang beban di wilayah yang sama dengan menggunakan konsol Elastic Load Balancing, CLI, atau API. Anda harus menyadari hal-hal berikut ini:

- Anda dapat melampirkan hanya satu penyeimbang beban ke lapisan.
- Setiap penyeimbang beban hanya dapat menangani satu lapisan.
- AWS OpsWorks Tumpukan tidak mendukung Application Load Balancer. Anda hanya dapat menggunakan Classic Load Balancer dengan AWS OpsWorks Stacks.

Ini berarti Anda harus membuat penyeimbang beban Elastic Load Balancing terpisah untuk setiap layer di setiap tumpukan yang ingin Anda seimbangkan dan gunakan hanya untuk tujuan itu. Praktik yang disarankan adalah menetapkan nama khusus untuk setiap penyeimbang beban Elastic Load Balancing yang Anda rencanakan untuk digunakan AWS OpsWorks dengan Stacks, MyStack seperti RailsLayer 1--ELB, untuk menghindari penggunaan penyeimbang beban untuk lebih dari satu tujuan.

> **Important**
>
> Sebaiknya buat load balancer Elastic Load Balancing baru untuk layer Stacks Anda. AWS OpsWorks Jika Anda memilih untuk menggunakan penyeimbang beban Elastic Load Balancing yang ada, Anda harus terlebih dahulu mengonfirmasi bahwa itu tidak digunakan untuk tujuan lain dan tidak memiliki instans yang terpasang. Setelah load balancer dilampirkan ke layer, OpsWorks menghapus instance yang ada dan mengonfigurasi load balancer untuk menangani hanya instance layer. Meskipun secara teknis dimungkinkan untuk menggunakan konsol Elastic Load Balancing atau API untuk memodifikasi konfigurasi load balancer setelah melampirkannya ke lapisan, Anda tidak boleh melakukannya; perubahan tidak akan permanen.

Untuk memasang penyeimbang beban Elastic Load Balancing ke lapisan

1. Jika Anda belum melakukannya, gunakan konsol Elastic Load Balancing, API, atau CLI untuk membuat penyeimbang beban di wilayah tumpukan. Saat Anda membuat penyeimbang beban, lakukan hal berikut:

   - Pastikan untuk menentukan jalur ping pemeriksaan kesehatan yang sesuai untuk aplikasi Anda.

     Jalur ping default adalah`/index.html`, jadi jika root aplikasi Anda tidak termasuk`index.html`, Anda harus menentukan jalur ping yang sesuai atau pemeriksaan kesehatan akan gagal.
   - Jika Anda ingin menggunakan pengurasan koneksi, pastikan fitur tersebut diaktifkan dan memiliki nilai batas waktu yang sesuai.

   Untuk informasi lebih lanjut, lihat Elastic Load Balancing.
2. Buat layer yang ingin Anda seimbangkan atau edit pengaturan Jaringan layer yang ada.

   > **Note**
   >
   > Anda tidak dapat melampirkan penyeimbang beban saat Anda membuat lapisan khusus. Anda harus mengedit pengaturan lapisan.

3. Di bawah Elastic Load Balancing, pilih penyeimbang beban yang ingin Anda lampirkan ke layer dan tentukan apakah Anda ingin AWS OpsWorks Stacks mendukung pengeringan koneksi.

Setelah Anda melampirkan penyeimbang beban ke lapisan, AWS OpsWorks Stacks memicu peristiwa siklus hidup Konfigurasi pada instance tumpukan untuk memberi tahu mereka tentang perubahan tersebut. AWS OpsWorks Tumpukan juga memicu peristiwa Konfigurasi saat Anda melepaskan penyeimbang beban.

> **Note**
>
> Setelah instance di-boot, AWS OpsWorks Stacks menjalankan resep Setup dan Deploy, yang menginstal paket dan menyebarkan aplikasi. Setelah resep tersebut selesai, instance dalam status online dan AWS OpsWorks Stacks mendaftarkan instance dengan Elastic Load Balancing. AWS OpsWorks Stacks juga memicu peristiwa Configure setelah instance online. Ini berarti bahwa pendaftaran Elastic Load Balancing dan resep Configure dapat berjalan secara bersamaan, dan instance mungkin terdaftar sebelum resep Configure selesai. Untuk memastikan bahwa resep selesai sebelum instance terdaftar dengan Elastic Load Balancing, Anda harus menambahkan resep ke peristiwa siklus hidup Setup atau Deploy layer. Untuk informasi selengkapnya, lihat Resep Eksekusi.

Terkadang berguna untuk menghapus instance dari penyeimbang beban. Misalnya, saat memperbarui aplikasi, sebaiknya Anda menerapkan aplikasi ke satu instance dan memverifikasi bahwa aplikasi berfungsi dengan baik sebelum menerapkannya ke setiap instance. Anda biasanya menghapus instance tersebut dari penyeimbang beban, sehingga instans tersebut tidak menerima permintaan pengguna hingga Anda memverifikasi pembaruan.

Anda harus menggunakan konsol Elastic Load Balancing atau API untuk menghapus instans online sementara dari penyeimbang beban. Berikut ini menjelaskan cara menggunakan konsol.

Untuk menghapus temporer instans dari penyeimbang beban

1. Buka EC2 konsol Amazon dan pilih Load Balancers.
2. Pilih penyeimbang beban yang sesuai dan buka tab Instances.
3. Pilih Hapus dari Load Balancer di kolom Actions instance.
4. Setelah selesai, pilih Edit Instans, dan kembalikan instans ke penyeimbang beban.

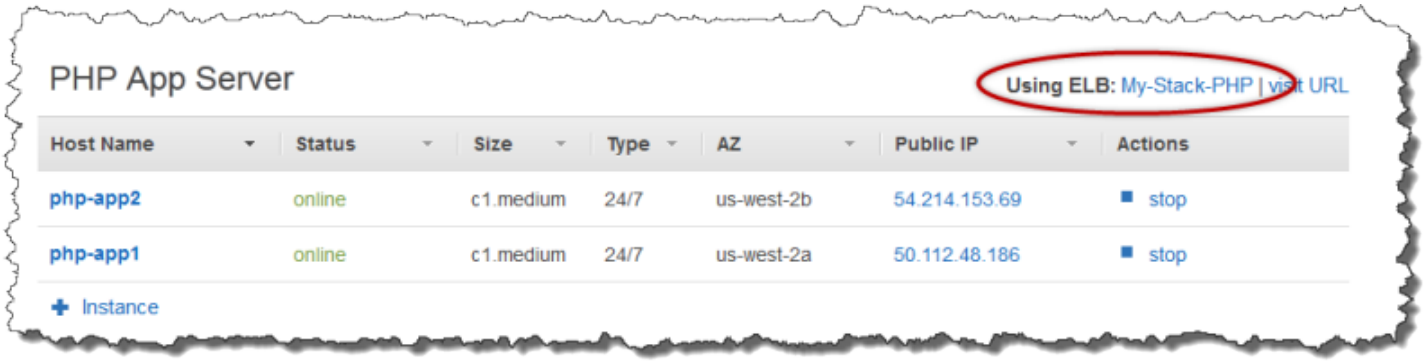
> **Important**
>
> Jika Anda menggunakan konsol Elastic Load Balancing atau API untuk menghapus instance dari load balancer, Anda juga harus menggunakan Elastic Load Balancing untuk mengembalikannya. AWS OpsWorks Stacks tidak mengetahui operasi yang Anda lakukan dengan konsol layanan lain atau APIs, dan itu tidak akan mengembalikan instance ke penyeimbang beban untuk Anda. Terkadang, AWS OpsWorks Stacks dapat menambahkan instance kembali ke ELB, tetapi ini bukan perilaku yang dijamin dan tidak terjadi di semua kasus.

Anda dapat melampirkan beberapa penyeimbang beban ke kumpulan instance tertentu sebagai berikut:

Untuk memasang beberapa penyeimbang beban

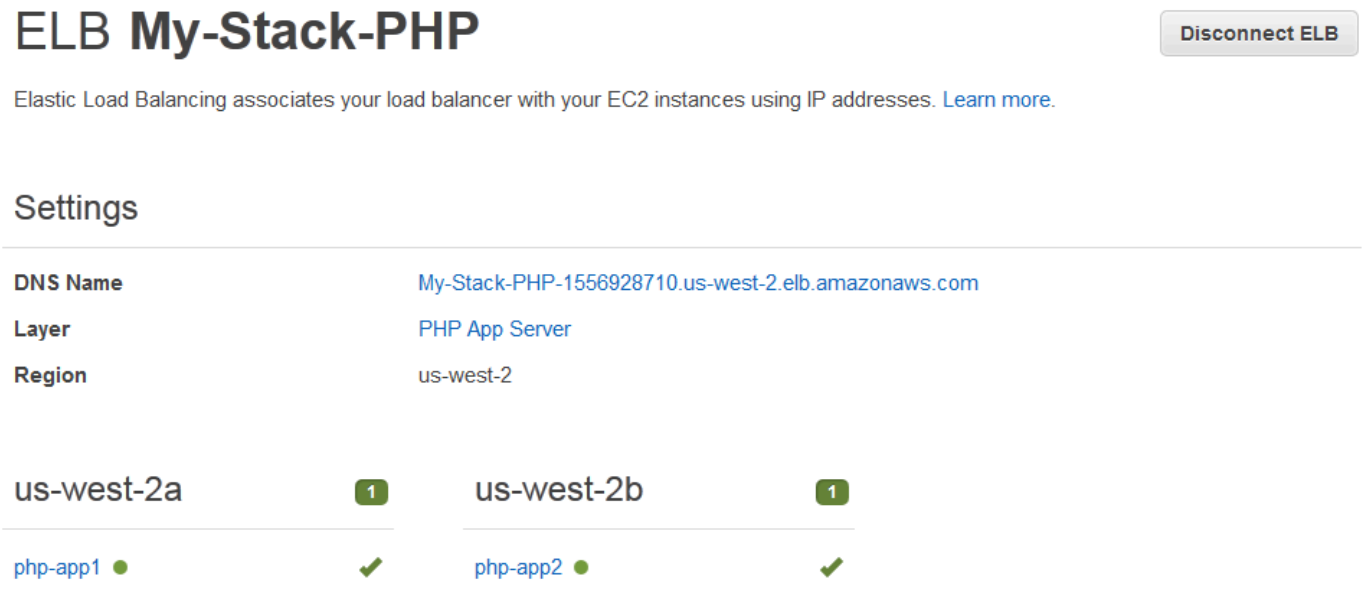
1. Gunakan konsol Elastic Load Balancing, API, atau CLI untuk membuat satu set penyeimbang beban.
2. Buat layer khusus untuk setiap penyeimbang beban dan pasang salah satu penyeimbang beban ke dalamnya. Anda tidak perlu menerapkan resep khusus apa pun untuk lapisan ini; lapisan kustom default sudah cukup.
3. Tambahkan kumpulan instance ke setiap lapisan kustom.

Anda dapat memeriksa properti penyeimbang beban dengan membuka halaman Instances dan mengklik nama penyeimbang beban yang sesuai.



Halaman ELB menunjukkan properti dasar penyeimbang beban, termasuk nama DNS dan status kesehatan instans terkait. Jika tumpukan berjalan di VPC, halaman menampilkan subnet daripada

Availibility Zones. Cek hijau menunjukkan contoh yang sehat. Anda dapat mengklik nama untuk terhubung ke server, melalui penyeimbang beban.



Lapisan Layanan Amazon RDS

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Lapisan layanan Amazon RDS mewakili instans Amazon RDS. Lapisan hanya dapat mewakili instans Amazon RDS yang ada, yang harus Anda buat secara terpisah dengan menggunakan konsol Amazon RDS atau API.

Prosedur dasar untuk memasukkan lapisan layanan Amazon RDS ke dalam tumpukan Anda adalah sebagai berikut:

1. Gunakan konsol Amazon RDS, API, atau CLI untuk membuat instans.

   Pastikan untuk merekam ID instans, nama pengguna master, kata sandi master, dan nama database.

2. Untuk menambahkan lapisan Amazon RDS ke tumpukan Anda, daftarkan instans Amazon RDS dengan tumpukan.
3. Lampirkan layer ke aplikasi, yang menambahkan informasi koneksi instans Amazon RDS ke `deploy`atribut aplikasi.
4. Gunakan file khusus bahasa atau informasi dalam `deploy` atribut untuk menghubungkan aplikasi ke instans Amazon RDS.

   Untuk informasi selengkapnya tentang cara menghubungkan aplikasi ke server database, lihat the section called “Menghubungkan ke Database”

> **Warning**
>
> Pastikan karakter dalam kata sandi utama dan nama pengguna instans kompatibel dengan server aplikasi Anda. Misalnya, dengan lapisan Java App Server, termasuk & dalam salah satu string menyebabkan kesalahan penguraian XMLyang mencegah server Tomcat memulai.

Topik

- Menentukan Grup Keamanan
- Mendaftarkan Instans Amazon RDS dengan Stack
- Mengaitkan Lapisan Layanan Amazon RDS dengan Aplikasi
- Menghapus Layer Layanan Amazon RDS dari Stack

Menentukan Grup Keamanan

Untuk menggunakan instans Amazon RDS dengan AWS OpsWorks Stacks, database atau grup keamanan VPC harus mengizinkan akses dari alamat IP yang sesuai. Untuk penggunaan produksi, grup keamanan biasanya membatasi akses hanya ke alamat IP yang perlu mengakses database. Ini biasanya mencakup alamat sistem yang Anda gunakan untuk mengelola database dan instance AWS OpsWorks Stacks yang perlu mengakses database. AWS OpsWorks Stacks secara otomatis membuat grup EC2 keamanan Amazon untuk setiap jenis lapisan saat Anda membuat tumpukan pertama di suatu wilayah. Cara sederhana untuk menyediakan akses untuk instans AWS OpsWorks Stacks adalah dengan menetapkan grup keamanan AWS OpsWorks Stacks yang sesuai ke instans Amazon RDS atau VPC.

Untuk menentukan grup keamanan untuk instans Amazon RDS yang ada

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Klik Instans di panel navigasi dan pilih instans Amazon RDS yang sesuai. Klik Tindakan Instance, Ubah.
3. Pilih grup keamanan berikut dari daftar Grup Keamanan dan kemudian klik Lanjutkan dan Ubah Instans DB untuk memperbarui instance.

   - Grup keamanan AWS- OpsWorks -DB-Master-Server (). *security_group_id*
   - Grup keamanan untuk lapisan server aplikasi yang instansnya akan terhubung ke database. Nama grup termasuk nama layer. Misalnya, untuk menyediakan akses database ke instance PHP App Server, tentukan grup AWS- OpsWorks -PHP-App-Server.

Jika Anda membuat instans Amazon RDS baru, Anda dapat menentukan grup keamanan AWS OpsWorks Stacks yang sesuai pada halaman Konfigurasi Pengaturan Lanjut panduan Launch DB Instance. Untuk deskripsi tentang cara menggunakan wizard ini, lihat Membuat Instans DB MySQL dan Menghubungkan ke Database pada Instans MySQL DB.

Untuk informasi tentang cara menentukan grup keamanan VPC, lihat Grup Keamanan untuk VPC Anda.

Mendaftarkan Instans Amazon RDS dengan Stack

Untuk menambahkan lapisan layanan Amazon RDS dalam tumpukan, Anda harus mendaftarkan instance dengan tumpukan.

Untuk mendaftarkan instans Amazon RDS dengan tumpukan

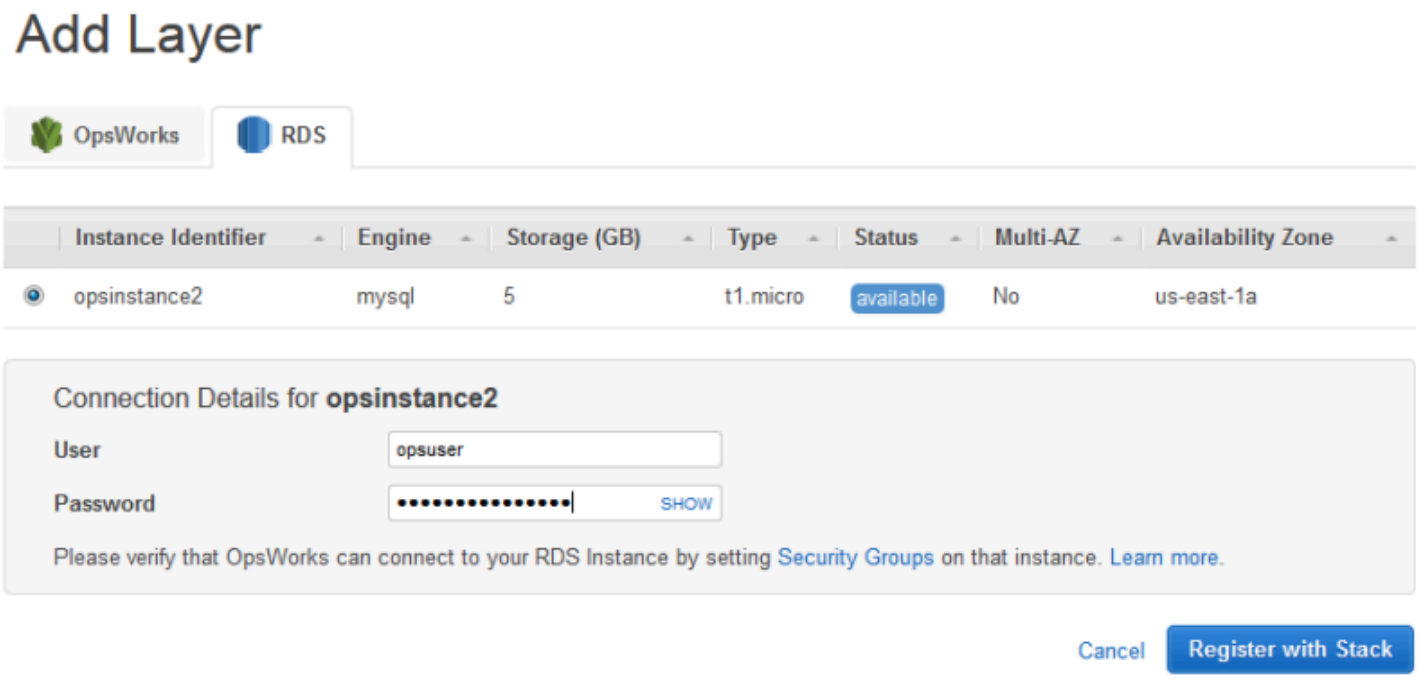
1. Di konsol AWS OpsWorks Stacks, klik Layer di panel navigasi, klik + Layer or Add a layer untuk membuka halaman Add Layer, dan kemudian klik tab RDS.
2. Jika perlu, perbarui peran layanan tumpukan, seperti yang dijelaskan dalamMemperbarui Peran Layanan Stack.
3. Klik tab RDS untuk mencantumkan instans Amazon RDS yang tersedia.

> **Note**
>
> Jika akun Anda tidak memiliki instans Amazon RDS, Anda dapat membuatnya dengan mengklik Tambahkan instans RDS di tab RDS, yang akan membawa Anda ke konsol

Amazon RDS dan memulai panduan Luncurkan Instans DB. Anda juga dapat langsung menuju konsol Amazon RDS dan klik Luncurkan Instans DB, atau gunakan Amazon RDS API atau CLI. Untuk informasi selengkapnya tentang cara membuat instans Amazon RDS, lihat Memulai dengan Amazon RDS.

4. Pilih contoh yang sesuai, atur Pengguna dan Kata Sandi ke nilai pengguna dan kata sandi yang sesuai dan klik Daftar ke Tumpukan.

> **Important**
>
> Anda harus memastikan bahwa pengguna dan kata sandi yang Anda gunakan untuk mendaftarkan instans Amazon RDS sesuai dengan pengguna dan kata sandi yang valid. Jika tidak, aplikasi Anda tidak akan dapat terhubung ke instance. Namun, Anda dapat mengedit layer untuk memberikan nilai pengguna dan kata sandi yang valid dan kemudian menerapkan ulang aplikasi.



Saat Anda menambahkan lapisan layanan Amazon RDS ke tumpukan, AWS OpsWorks Stacks memberinya ID dan menambahkan konfigurasi Amazon RDS terkait ke atribut atribut konfigurasi stack dan deployment. `[:opsworks][:stack]`

> **Note**
>
> Jika Anda mengubah kata sandi instans Amazon RDS terdaftar, Anda harus memperbarui kata sandi secara manual di AWS OpsWorks Stacks dan kemudian menerapkan ulang aplikasi Anda untuk memperbarui konfigurasi tumpukan dan atribut penerapan pada instance tumpukan.

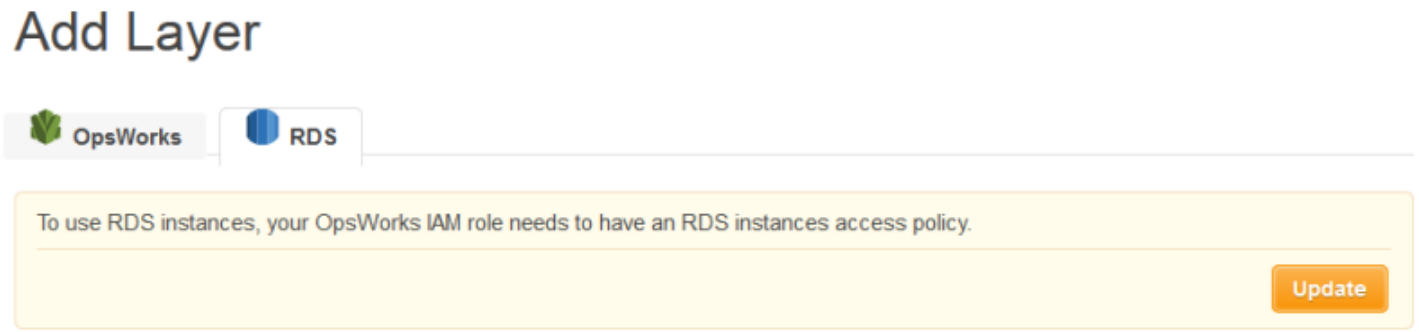
Topik

- Memperbarui Peran Layanan Stack

Memperbarui Peran Layanan Stack

Setiap tumpukan memiliki peran layanan IAM yang menentukan tindakan apa yang dapat dilakukan AWS OpsWorks Stacks atas nama Anda dengan layanan AWS lainnya. Untuk mendaftarkan instans Amazon RDS dengan tumpukan, peran layanannya harus memberikan izin AWS OpsWorks Stacks untuk mengakses Amazon RDS.

Saat pertama kali Anda menambahkan lapisan layanan Amazon RDS ke salah satu tumpukan Anda, peran layanan mungkin tidak memiliki izin yang diperlukan. Jika demikian, ketika Anda mengklik tab RDS pada halaman Add Layer, Anda akan melihat yang berikut ini.



Klik Perbarui agar AWS OpsWorks Tumpukan memperbarui kebijakan peran layanan ke yang berikut ini.

```
{"Statement": [{"Action": ["ec2:*", "iam:PassRole",
                      "cloudwatch:GetMetricStatistics",
                      "elasticloadbalancing:*",
                      "rds:*"],
           "Effect": "Allow",
```

```
            "Resource": ["*"] }]
}
```

> **Note**
>
> Anda hanya perlu melakukan pembaruan sekali. Peran yang diperbarui kemudian secara otomatis digunakan oleh semua tumpukan Anda.

Mengaitkan Lapisan Layanan Amazon RDS dengan Aplikasi

Setelah menambahkan lapisan layanan Amazon RDS, Anda dapat mengaitkannya dengan aplikasi.

- Anda dapat mengaitkan lapisan Amazon RDS ke aplikasi saat membuat aplikasi, atau yang lebih baru dengan mengedit konfigurasi aplikasi.
- Untuk memisahkan lapisan Amazon RDS dari aplikasi, edit konfigurasi aplikasi untuk menentukan server database yang berbeda, atau tidak ada server.

  Lapisan Amazon RDS tetap menjadi bagian dari tumpukan, dan dapat dikaitkan dengan aplikasi yang berbeda.

Setelah Anda mengaitkan instans Amazon RDS dengan aplikasi, AWS OpsWorks Stacks menempatkan informasi koneksi database di server aplikasi. Aplikasi pada setiap instance server kemudian dapat menggunakan informasi ini untuk terhubung ke database. Untuk informasi selengkapnya tentang cara menyambung ke instans Amazon RDS, lihatthe section called “Menghubungkan ke Database”.

Menghapus Layer Layanan Amazon RDS dari Stack

Untuk menghapus lapisan layanan Amazon RDS dari tumpukan, Anda membatalkan pendaftarannya.

Untuk membatalkan pendaftaran lapisan layanan Amazon RDS

1. Klik Layers di panel navigasi dan klik nama layer layanan Amazon RDS.
2. Klik Deregister dan konfirmasikan bahwa Anda ingin membatalkan pendaftaran layer.

Prosedur ini menghapus lapisan dari tumpukan, tetapi tidak menghapus instance Amazon RDS yang mendasarinya. Instans dan database apa pun tetap ada di akun Anda dan dapat didaftarkan dengan

tumpukan lain. Anda harus menggunakan konsol Amazon RDS, API, atau CLI untuk menghapus instance. Untuk informasi lebih lanjut, lihat Menghapus instans DB.

Lapisan Klaster ECS

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Layanan Amazon Elastic Container Service (Amazon ECS) mengelola kontainer Docker pada sekelompok instans Amazon Elastic Compute Cloud (Amazon EC2), yang dikenal sebagai instance container. Lapisan ECS Cluster mewakili kluster Amazon ECS, dan menyederhanakan pengelolaan klaster dengan menyediakan fitur yang mencakup:

- Penyediaan dan manajemen instance kontainer yang efisien
- Sistem operasi instans kontainer dan pembaruan paket
- Manajemen izin pengguna
- Pemantau performa instans
- Manajemen volume Amazon Elastic Block Store (Amazon EBS)
- Manajemen alamat IP Elastic IP
- Manajemen grup keamanan

Lapisan ECS Cluster memiliki batasan dan persyaratan berikut:

- Lapisan ini hanya tersedia untuk tumpukan Chef 11.10 atau Chef 12 Linux yang berjalan di VPC, termasuk VPC default.
- Instans lapisan harus menjalankan salah satu sistem operasi berikut.
  - 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
- Kustom

- Versi agen AWS OpsWorks Stacks pada instance layer harus `3425-20150727112318` atau lebih baru.

Topik

- Menambahkan Layer Cluster ECS ke Stack
- Mengelola Cluster ECS
- Menghapus Layer Cluster ECS dari Stack

Menambahkan Layer Cluster ECS ke Stack

AWS OpsWorks Stacks menyederhanakan proses peluncuran dan pemeliharaan instans kontainer untuk kluster Amazon ECS yang ada. Untuk membuat atau meluncurkan entitas Amazon ECS lainnya, seperti cluster dan tugas, gunakan konsol Amazon ECS, antarmuka baris perintah (CLI), atau API. (Untuk informasi selengkapnya, lihat Panduan Pengembang Amazon Elastic Container Service Developer.) Anda kemudian dapat mengaitkan cluster dengan tumpukan dengan membuat lapisan ECS Cluster, yang dapat Anda gunakan untuk mengelola cluster di AWS OpsWorks Stacks.

Anda dapat mengaitkan cluster dengan tumpukan sebagai berikut:

- Setiap tumpukan dapat memiliki satu lapisan ECS Cluster, yang mewakili satu cluster.
- Klaster dapat dikaitkan dengan hanya satu tumpukan.

Sebelum Anda dapat menambahkan lapisan ECS Cluster ke tumpukan Anda, Anda harus memperbarui peran layanan AWS OpsWorks Stacks AWS Identity and Access Management (IAM), yang biasanya diberi nama`aws-opsworks-service-role`, untuk memungkinkan AWS OpsWorks

Stacks berinteraksi dengan Amazon ECS atas nama Anda. Untuk informasi selengkapnya tentang peran layanan, lihatMengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda.

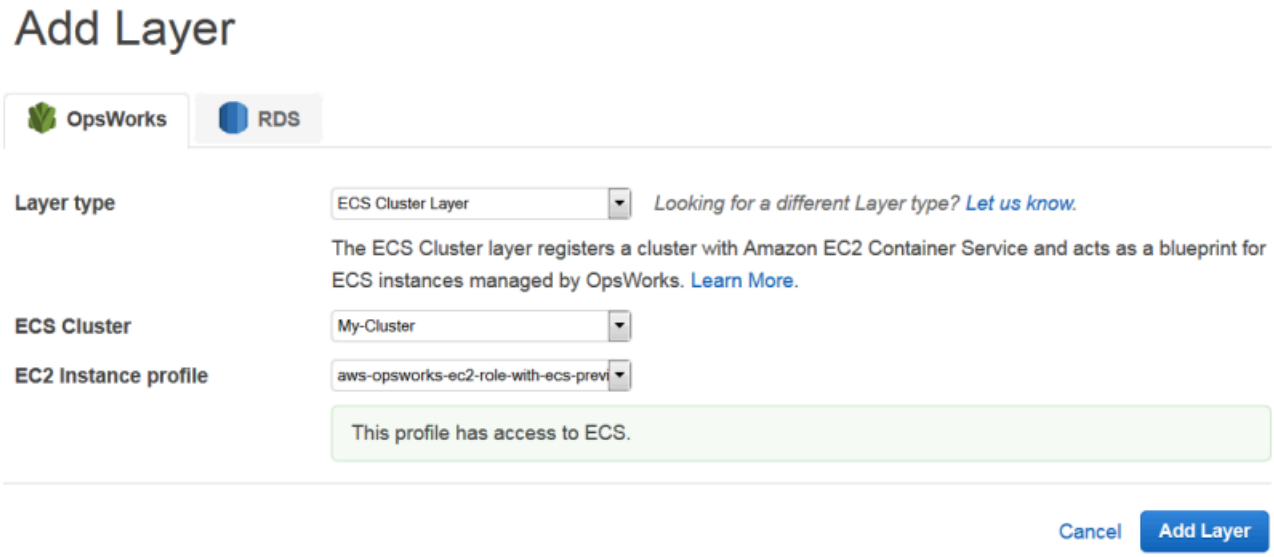
Pertama kali Anda membuat layer ECS Cluster, konsol menyediakan tombol Update yang dapat Anda pilih untuk mengarahkan AWS OpsWorks Stacks untuk memperbarui peran untuk Anda. AWS OpsWorks Stacks kemudian menampilkan halaman Add Layer sehingga Anda dapat menambahkan layer ke stack. Anda perlu memperbarui peran layanan hanya sekali. Anda kemudian dapat menggunakan peran yang diperbarui untuk menambahkan lapisan ECS Cluster ke tumpukan apa pun.

> **Note**
>
> Jika mau, Anda dapat memperbarui kebijakan peran layanan secara manual dengan menambahkan `ecs:*` izin ke kebijakan yang ada, sebagai berikut:

```
{
  "Statement": [
    {
      "Action": [
        "ec2:*",
        "iam:PassRole",
        "cloudwatch:GetMetricStatistics",
        "elasticloadbalancing:*",
        "rds:*",
        "ecs:*"
      ],
      "Effect": "Allow",
      "Resource": ["*"]
    }
  ]
}
```

Mengaitkan cluster dengan tumpukan membutuhkan dua operasi: mendaftarkan cluster dengan tumpukan dan kemudian membuat lapisan terkait. Konsol AWS OpsWorks Stacks menggabungkan langkah-langkah ini; pembuatan lapisan secara otomatis mendaftarkan cluster yang ditentukan. Jika Anda menggunakan AWS OpsWorks Stacks API, CLI, atau SDK, Anda harus menggunakan operasi terpisah untuk mendaftarkan cluster dan membuat layer terkait. Untuk menggunakan konsol untuk

menambahkan layer ECS Cluster ke stack Anda, pilih Layers, pilih +Layer atau Add a Layer, lalu pilih tipe layer ECS Cluster.



Halaman Add Layer mencakup opsi konfigurasi berikut:

Klaster ECS

Cluster Amazon ECS yang ingin Anda daftarkan dengan tumpukan.

EC2 Profil instans

Profil instans Amazon Elastic Compute Cloud (Amazon EC2) klaster. Profil ini memberikan izin untuk aplikasi yang berjalan pada instans penampung klaster untuk mengakses layanan AWS lainnya, termasuk Amazon ECS. Saat Anda membuat layer ECS Cluster pertama Anda, pilih Profil baru dengan akses ECS ke AWS OpsWorks Stacks langsung untuk membuat profil yang diperlukan, yang diberi nama. `aws-opsworks-ec2-role-with-ecs` Anda kemudian dapat menggunakan profil itu untuk semua lapisan ECS Cluster berikutnya. Untuk informasi lebih lanjut tentang profil instans, lihatMenentukan Izin untuk Aplikasi yang Berjalan pada instance EC2 .

Anda dapat menentukan pengaturan lain dengan mengedit konfigurasi lapisan, termasuk:

- Melampirkan penyeimbang beban Elastic Load Balancing ke lapisan.

  Pendekatan ini mungkin cocok untuk beberapa kasus penggunaan, tetapi Amazon ECS menyediakan opsi yang lebih canggih. Untuk informasi lebih lanjut, lihat Penyeimbang Beban Layanan.

- Menentukan apakah akan secara otomatis menetapkan alamat IP publik atau alamat IP Elastis ke instance container.

  Jika Anda menonaktifkan penugasan otomatis untuk kedua jenis alamat, instance tidak akan online kecuali subnet memiliki NAT yang dikonfigurasi dengan benar. Untuk informasi selengkapnya, lihat Menjalankan Stack di VPC.

Mengelola Cluster ECS

Setelah Anda membuat layer ECS Cluster, Anda dapat menggunakan AWS OpsWorks Stacks untuk mengelola cluster sebagai berikut:

Menyediakan dan mengelola instans kontainer

Awalnya, lapisan ECS Cluster tidak menyertakan instance kontainer apa pun, bahkan jika cluster asli melakukannya. Salah satu opsi adalah mengelola instance layer dengan menggunakan kombinasi yang sesuai dari berikut ini:

- Tambahkan instance 24/7 secara manual ke layer dan hapus saat tidak lagi diperlukan.
- Menambahkan atau menghapus instance pada jadwal dengan menambahkan instance berbasis waktu ke layer.
- Menambahkan atau menghapus instance berdasarkan metrik host AWS OpsWorks Stacks atau CloudWatch alarm dengan menambahkan instance berbasis beban ke layer.

> **Note**
>
> Jika Amazon ECS tidak didukung untuk sistem operasi default stack, Anda harus secara eksplisit menentukan sistem operasi yang didukung—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 04 LTS, atau kustom—saat Anda membuat instance kontainer. Jangan gunakan AMI yang Dioptimalkan ECS untuk membuat instance di lapisan ECS, karena AMI ini sudah menyertakan agen ECS. AWS OpsWorks Stacks juga mencoba menginstal agen ECS selama proses penyiapan instans, dan konflik dapat menyebabkan penyiapan gagal.

Untuk informasi selengkapnya, lihatMengoptimalkan Jumlah Server. AWS OpsWorks Stacks menetapkan grup keamanan AWS- OpsWorks -ECS-Cluster ke setiap instance. Setelah setiap

instance baru selesai booting, AWS OpsWorks Stacks mengubahnya menjadi instance container dengan menginstal Docker dan agen Amazon ECS, lalu mendaftarkan instance tersebut ke cluster.

Jika Anda lebih suka menggunakan instance kontainer yang ada, Anda dapat mendaftarkannya dengan tumpukan dan menetapkannya ke lapisan ECS Cluster. Perhatikan bahwa instance harus menjalankan sistem operasi yang didukung, Amazon Linux 2015.03 atau yang lebih baru, atau Ubuntu 14.04 LTS atau yang lebih baru.

> **Note**
>
> Instance kontainer tidak dapat dimiliki oleh lapisan ECS Cluster dan lapisan bawaan lainnya. Namun, instance kontainer dapat menjadi milik lapisan ECS Cluster dan satu atau lebih lapisan khusus.

Jalankan pembaruan sistem operasi dan paket

Setelah instance baru selesai booting, AWS OpsWorks Stacks menginstal pembaruan terbaru. Anda kemudian dapat menggunakan AWS OpsWorks Stacks untuk menjaga instance kontainer tetap mutakhir. Untuk informasi selengkapnya, lihat Mengelola Pembaruan Keamanan.

Mengelola izin pengguna

AWS OpsWorks Stacks menyediakan cara sederhana untuk mengelola izin pada instance container, termasuk mengelola kunci SSH pengguna. Untuk informasi selengkapnya, lihat Mengelola izin dan Mengelola Akses SSH.

Memantau metrik kinerja

AWS OpsWorks Stacks menyediakan berbagai cara untuk memantau metrik kinerja untuk stack, layer, atau instance individual. Untuk informasi selengkapnya, lihat Pemantauan.

Anda menangani tugas manajemen lainnya, seperti membuat tugas atau layanan, melalui Amazon ECS. Untuk informasi selengkapnya, silakan lihat Panduan Developer Amazon Elastic Container Service.

> **Note**
>
> Untuk langsung menuju ke halaman cluster di konsol Amazon ECS, pilih Instances, lalu pilih ECS Cluster, yang berada di dekat sudut kanan atas bagian lapisan ECS Cluster.

Menghapus Layer Cluster ECS dari Stack

Bila Anda tidak lagi membutuhkan cluster, hapus layer ECS Cluster dan deregister cluster terkait. Menghapus cluster dari tumpukan membutuhkan dua operasi: membatalkan pendaftaran cluster dan kemudian menghapus lapisan terkait. Konsol AWS OpsWorks Stacks menggabungkan langkah-langkah ini; penghapusan lapisan secara otomatis membatalkan pendaftaran cluster yang ditentukan. Jika Anda menggunakan AWS OpsWorks Stacks API, CLI, atau SDK, Anda harus menggunakan operasi terpisah untuk membatalkan pendaftaran cluster dan menghapus lapisan terkait.

Untuk menggunakan konsol untuk menghapus lapisan ECS Cluster

1. Jika Anda ingin mengontrol cara tugas dimatikan, gunakan konsol Amazon ECS, API, atau CLI untuk mengurangi dan menghapus layanan klaster. Untuk informasi selengkapnya, lihat Membersihkan Sumber Daya Amazon ECS Anda.
2. Hentikan instance layer, lalu hapus. Saat Anda menghentikan instance kontainer, AWS OpsWorks Stacks secara otomatis menghentikan tugas apa pun yang sedang berjalan, membatalkan pendaftaran instance dari cluster, dan menghentikan instance.

   > **Note**
   >
   > Jika Anda telah mendaftarkan instance kontainer yang ada dengan tumpukan, Anda dapat membatalkan penetapan instance dari lapisan dan kemudian membatalkan pendaftarannya, yang mengembalikan instance ke kontrol ECS.

3. Hapus layer. AWS OpsWorks Tumpukan membatalkan pendaftaran cluster terkait, tetapi tidak menghapusnya. Cluster tetap di Amazon ECS.

Lapisan AWS OpsWorks Tumpukan Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Lapisan khusus hanya memiliki satu set resep minimal. Anda kemudian menambahkan fungsionalitas yang sesuai ke lapisan dengan menerapkan resep khusus dan menetapkannya ke peristiwa siklus hidup lapisan.

Lapisan kustom memiliki pengaturan konfigurasi berikut.

> **Note**
>
> AWS OpsWorks Stacks secara otomatis menginstal Ruby pada instance layer. Jika Anda ingin menjalankan kode Ruby pada instance tetapi tidak ingin menggunakan versi Ruby default, Anda dapat menggunakan JSON kustom atau file atribut khusus untuk menentukan versi pilihan Anda. Untuk informasi selengkapnya, lihat Versi Ruby.

Prosedur dasar untuk membuat lapisan kustom memiliki langkah-langkah berikut:

1. Menerapkan buku masak yang berisi resep dan file terkait yang diperlukan untuk menginstal dan mengonfigurasi paket, menangani perubahan konfigurasi, menyebarkan aplikasi, dan sebagainya.

   Bergantung pada kebutuhan Anda, Anda mungkin juga memerlukan resep untuk menangani tugas undeployment dan shutdown. Untuk informasi selengkapnya, lihat Buku Masak dan Resep.
2. Buat layer kustom.
3. Tetapkan resep Anda ke acara siklus hidup yang sesuai.

Anda kemudian menambahkan instance ke layer, memulainya, dan menerapkan aplikasi ke instance tersebut.

> **Important**
>
> Untuk menerapkan aplikasi ke instance lapisan kustom, Anda harus menerapkan resep untuk menangani operasi penerapan dan menetapkannya ke acara Deploy layer.

Instalasi Paket Sistem Operasi Per-layer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Dimulai dengan Chef 12, Anda harus menggunakan resep khusus untuk menginstal paket pada lapisan yang menjalankan sistem operasi yang berbeda. Pendekatan ini memberi Anda fleksibilitas dan kontrol maksimum atas instalasi paket.

Misalnya, misalkan Anda ingin menginstal Apache pada lapisan yang sedang berjalan RedHat, Ubuntu, dan versi Amazon dari sistem operasi Linux. Paket Apache untuk RedHat dan Amazon Linux disebut`httpd`, tetapi di Ubuntu, disebut`apache2`.

Untuk mengatasi perbedaan dalam penamaan paket, Anda dapat menggunakan sintaks yang mirip dengan yang ada di resep contoh berikut. Resepnya menginstal paket Apache yang sesuai untuk setiap sistem operasi. Contoh ini didasarkan pada dokumentasi Chef.

```
package "Install Apache" do
   case node[:platform]
      when "redhat", "amazon"
         package_name "httpd"
      when "ubuntu"
         package_name "apache2"
   end
end
```

Untuk informasi rinci tentang cara menggunakan `package` sumber daya untuk mengelola paket, buka halaman paket di dokumentasi Chef.

Atau, Anda dapat menggunakan metode `value_for_platform` pembantu dari Chef Recipe DSL (bahasa khusus domain), yang menyelesaikan hal yang sama secara lebih ringkas:

```
package "Install Apache" do
   package_name value_for_platform(
      ["redhat", "amazon"] => { "default" => "httpd" },
      ["ubuntu"] => { "default" => "apache2" }
   )
end
```

Untuk informasi tentang penggunaan metode `value_for_platform` pembantu, buka Tentang Resep DSL.

Instans

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Instance mewakili sumber daya komputasi, seperti EC2 instans Amazon, yang menangani pekerjaan melayani aplikasi, menyeimbangkan lalu lintas, dan sebagainya. Sistem operasi instans dapat memiliki salah satu dari beberapa distribusi Linux, atau Windows Server 2012 R2.

Anda dapat menambahkan instance ke tumpukan dengan salah satu cara berikut:

- Gunakan AWS OpsWorks Stacks untuk menambahkan instance ke tumpukan. Instance yang Anda tambahkan mewakili EC2 instans Amazon.
- Untuk tumpukan berbasis Linux, Anda dapat mendaftarkan instance yang dibuat di tempat lain—termasuk instance yang dibuat dengan EC2 Amazon dan instans lokal yang berjalan di perangkat keras Anda sendiri.

Anda kemudian dapat menggunakan AWS OpsWorks Stacks untuk mengelola instance ini dengan cara yang hampir sama seperti instance yang dibuat dengan Stacks AWS OpsWorks

Bagian ini menjelaskan cara menggunakan AWS OpsWorks Stacks untuk membuat dan mengelola instance.

Topik

- Menggunakan AWS OpsWorks Instans Stacks
- Menggunakan Sumber Daya Komputasi yang Dibuat Di Luar AWS OpsWorks Tumpukan
- Mengedit Konfigurasi Instance
- Menghapus Instans AWS OpsWorks Stacks
- Menggunakan SSH untuk Masuk ke Instance Linux
- Menggunakan RDP untuk Masuk ke Instans Windows

Menggunakan AWS OpsWorks Instans Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menggunakan AWS OpsWorks Stacks untuk membuat instance dan menambahkannya ke tumpukan.

Topik

- AWS OpsWorks Stacks sistem operasi
- Menambahkan Instance ke Layer
- Menggunakan Custom AMIs
- Memulai, Menghentikan, dan Memulai Ulang Instans 24/7 Secara Manual
- Mengelola beban dengan instans berbasis waktu dan berbasis beban

AWS OpsWorks Stacks sistem operasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks mendukung versi 64-bit dari beberapa sistem operasi bawaan, termasuk distribusi Amazon dan Ubuntu Linux, dan Microsoft Windows Server. Beberapa catatan umum:

- Instans tumpukan dapat menjalankan Linux atau Windows.

  Tumpukan dapat memiliki versi atau distribusi Linux yang berbeda pada instance yang berbeda, tetapi Anda tidak dapat mencampur instance Linux dan Windows.
- Anda dapat menggunakan kustom AMIs (Amazon Machine Images), tetapi mereka harus didasarkan pada salah satu AWS OpsWorks Stacks-support AMIs yang dijelaskan dalam topik di bagian ini. Meskipun dimungkinkan untuk membuat atau mendaftarkan instance dengan sistem operasi lain (seperti CentOS 6. x) yang telah dibuat dari kustom atau yang dihasilkan oleh komunitas AMIs, ini tidak didukung secara resmi.
  - Sistem operasi Linux
  - Microsoft Windows Server
- Anda dapat memulai dan menghentikan instance secara manual atau membuat AWS OpsWorks Stacks secara otomatis menskalakan jumlah instance.

  Anda dapat menggunakan penskalaan otomatis berbasis waktu dengan tumpukan apa pun; Tumpukan Linux juga dapat menggunakan penskalaan berbasis beban.
- Selain menggunakan AWS OpsWorks Stacks untuk membuat EC2 instans Amazon, Anda juga dapat mendaftarkan instance dengan tumpukan Linux yang dibuat di luar Stacks. AWS OpsWorks

  Ini termasuk instans dan EC2 instans Amazon yang berjalan di perangkat keras Anda sendiri. Namun, mereka harus menjalankan salah satu distribusi Linux yang didukung. Anda tidak dapat mendaftarkan instans Amazon EC2 atau Windows lokal.

Anda dapat menjalankan AWS OpsWorks Stacks DescribeOperatingSystemsAPI untuk menampilkan daftar sistem operasi yang didukung dan versi Chef yang didukung. Berikut ini adalah contoh perintah, menggunakan AWS CLI.

```
aws opsworks describe-operating-systems
```

Berikut ini adalah contoh respons.

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

Topik

- Sistem operasi Linux
- Microsoft Windows Server

Sistem operasi Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks mendukung versi 64-bit dari sistem operasi Linux berikut.

- Amazon Linux dan Amazon Linux 2 (lihat konsol AWS OpsWorks Stacks untuk versi yang saat ini didukung)
- Ubuntu 20.04 LTS
- CentOS 7

- Perusahaan Topi Merah Linux 7

Anda juga dapat menggunakan kustom AMIs berdasarkan sistem operasi ini.

Beberapa catatan umum tentang instance Linux:

Versi paket yang didukung

Versi yang didukung dan tingkat patch untuk paket, seperti Ruby, bergantung pada sistem operasi dan versi seperti yang dijelaskan di bagian berikut.

Pembaruan

Secara default, AWS OpsWorks Stacks memastikan bahwa instance Linux memiliki patch keamanan terbaru dengan secara otomatis memanggil `yum update` atau `apt-get update` setelah instance boot. Untuk menonaktifkan pembaruan otomatis, gunakan CreateInstance, UpdateInstanceCreateLayer, atau UpdateLayertindakan—atau metode AWS SDK yang setara atau perintah AWS CLI — untuk menyetel parameter ke. `InstallUpdatesOnBoot` `false`

Untuk menghindari gangguan layanan, AWS OpsWorks Stacks tidak secara otomatis menginstal pembaruan setelah instans online. Anda dapat memperbarui sistem operasi instans online secara manual kapan saja dengan menjalankan perintah tumpukan Sistem Operasi Upgrade. Untuk informasi selengkapnya tentang cara mengelola pembaruan keamanan, lihatMengelola Pembaruan Keamanan.

Untuk kontrol lebih lanjut tentang cara AWS OpsWorks Stacks memperbarui instans Anda, buat AMI kustom berdasarkan salah satu sistem operasi yang didukung. Misalnya, dengan kustom AMIs Anda dapat menentukan versi paket mana yang diinstal pada sebuah instance. Setiap distribusi Linux memiliki garis waktu dukungan dan kebijakan penggabungan paket yang berbeda, jadi Anda harus mempertimbangkan pendekatan mana yang paling sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya, lihat Menggunakan Custom AMIs.

Berkas host

Setiap instance online memiliki `/etc/hosts` file yang memetakan alamat IP ke nama host. AWS OpsWorks Stacks mencakup alamat publik dan pribadi untuk semua instance online stack di setiap file instance. `hosts` Misalnya, Anda memiliki tumpukan dengan dua instance Server Aplikasi Node.js, nodejs-app1 dan nodejs-app2, dan satu instance MySQL, db-master1. `hosts`File instance nodejs-app1 akan terlihat seperti contoh berikut, dan instance lain akan memiliki file serupa. `hosts`

```
...
# 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 Dukungan proxy agen Stacks

Agen AWS OpsWorks Stacks untuk Chef 11.10 dan tumpukan yang lebih baru mencakup dukungan dasar untuk server proxy, yang biasanya digunakan dengan terisolasi. VPCs Untuk mengaktifkan dukungan server proxy, sebuah instance harus memiliki `/etc/environment` file yang menyediakan pengaturan yang sesuai untuk lalu lintas HTTP dan HTTPS. File akan terlihat mirip dengan yang berikut ini, di mana Anda mengganti teks yang disorot dengan URL dan port server proxy Anda:

```
http_proxy="http://myproxy.example.com:8080/"
https_proxy="http://myproxy.example.com:8080/"
no_proxy="169.254.169.254"
```

Untuk mengaktifkan dukungan proxy, sebaiknya buat AMI kustom yang menyertakan `/etc/environment` file yang sesuai dan menggunakan AMI tersebut untuk membuat instance Anda.

> **Note**
>
> Kami tidak menyarankan menggunakan resep khusus untuk membuat `/etc/environment` file pada instance Anda. AWS OpsWorks Tumpukan membutuhkan data server proxy di awal proses penyiapan, sebelum resep khusus apa pun dijalankan.

Topik

- Amazon Linux
- Ubuntu LTS
- CentOS

- Linux Red Hat Enterprise

Amazon Linux

AWS OpsWorks Stacks mendukung versi 64-bit Amazon Linux dan Amazon Linux 2. Selain pembaruan dan tambalan rutin, Amazon Linux merilis versi baru kira-kira setiap enam bulan, yang dapat melibatkan perubahan signifikan. Saat Anda membuat tumpukan atau instance baru, Anda harus menentukan versi Amazon Linux mana yang akan digunakan. Saat AWS merilis versi baru, instans Anda akan terus menjalankan versi yang ditentukan hingga Anda mengubahnya secara eksplisit. Setelah versi Amazon Linux baru dirilis, ada periode migrasi empat minggu, di mana AWS terus memberikan pembaruan rutin untuk versi lama. Setelah periode migrasi berakhir, instans Anda dapat terus menjalankan versi lama, tetapi AWS tidak menyediakan pembaruan lebih lanjut. Untuk informasi lebih lanjut, lihat Amazon Linux AMI FAQs.

Ketika versi Amazon Linux baru dirilis, kami sarankan Anda memperbarui ke versi baru dalam periode migrasi sehingga instans Anda terus menerima pembaruan keamanan. Sebelum memperbarui instance tumpukan produksi, sebaiknya Anda memulai instance baru dan memverifikasi bahwa aplikasi Anda berjalan dengan benar di versi baru. Anda kemudian dapat memperbarui instance tumpukan produksi.

> **Note**
>
> Secara default, kustom AMIs berbasis Amazon Linux secara otomatis diperbarui ke versi baru ketika dirilis. Praktik yang disarankan adalah mengunci AMI kustom Anda ke versi Amazon Linux tertentu sehingga Anda dapat menunda pembaruan hingga Anda menguji versi baru. Untuk informasi selengkapnya, lihat Bagaimana cara mengunci AMI ke versi tertentu? .
> Jika Anda menggunakan AWS CloudFormation template untuk membuat tumpukan dengan instance yang menjalankan Amazon Linux, template harus secara eksplisit menentukan versi Amazon Linux. Secara khusus, jika template Anda menentukan`Amazon Linux`, instance akan terus menjalankan versi 2016.09. Untuk informasi selengkapnya, silakan lihat AWS::OpsWorks::Stack dan AWS::OpsWorks::Instance.

Untuk memperbarui versi Amazon Linux instans, lakukan salah satu hal berikut:

- Untuk instance online, jalankan perintah tumpukan Sistem Operasi Upgrade.

Ketika versi Amazon Linux baru tersedia, halaman Instances dan Stack menampilkan pemberitahuan dengan tautan yang membawa Anda ke halaman Run Command. Anda kemudian dapat menjalankan Upgrade Sistem Operasi untuk meng-upgrade instance Anda.

- Untuk instans offline Amazon Elastic Block Store (didukung EBS), mulai instans dan jalankan Sistem Operasi Upgrade, seperti yang dijelaskan dalam pernyataan sebelumnya.
- Untuk instans yang didukung penyimpanan instans offline, termasuk instance berbasis waktu dan berbasis beban, edit setelan sistem Operasi instans untuk menentukan versi baru.

  AWS OpsWorks Stacks secara otomatis memperbarui instance ke versi baru saat dimulai ulang.

Amazon Linux: Versi Node.js yang Didukung

Versi Amazon Linux	Versi 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 0.12.13

Versi Amazon Linux	Versi Node.js
	0.12.12 0.12.10

Amazon Linux: Versi Chef yang Didukung

Versi Koki	Versi Amazon Linux yang Didukung
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

Sebelum memperbarui instance t1.micro, pastikan mereka memiliki file swap sementara,. `/var/swapfile` Instans t1.micro pada tumpukan Chef 0.9 tidak memiliki file swap. Untuk tumpukan Chef 11.4 dan Chef 11.10, versi terbaru agen instans secara otomatis membuat file swap untuk instans t1.micro. Namun, perubahan ini diperkenalkan selama beberapa minggu, jadi Anda harus memeriksa keberadaan `/var/swapfile` instance yang dibuat sebelum sekitar 24 Maret 2014.

Untuk instance t1.micro yang tidak memiliki file swap, Anda dapat membuatnya sebagai berikut:

- Untuk tumpukan Chef 11.10 dan yang lebih baru, buat instance t1.micro baru, yang secara otomatis memiliki file swap.
- Untuk tumpukan Chef 0.9, jalankan perintah berikut pada setiap instance sebagai pengguna root.

```
dd if=/dev/zero of=/var/swapfile bs=1M count=256
 mkswap /var/swapfile
 chown root:root /var/swapfile
 chmod 0600 /var/swapfile
 swapon /var/swapfile
```

Anda juga dapat menggunakan perintah ini di Chef 11.10 dan tumpukan yang lebih baru jika Anda tidak ingin membuat instance baru.

Ubuntu LTS

Ubuntu merilis versi Ubuntu LTS baru kira-kira setiap dua tahun dan mendukung setiap rilis selama kurang lebih lima tahun. Ubuntu menyediakan patch keamanan dan update selama durasi dukungan sistem operasi. Untuk informasi selengkapnya, lihat LTS - Ubuntu Wiki.

- Anda tidak dapat memperbarui instance Ubuntu yang ada ke rilis Ubuntu yang lebih baru.

  Anda harus membuat instance Ubuntu baru dan menghapus instance lama.
- Ubuntu 20.04 LTS hanya didukung untuk Chef 12 dan tumpukan yang lebih tinggi.

CentOS

AWS OpsWorks Stacks mendukung CentOS 7 versi 64-bit. Versi awal yang didukung adalah CentOS 7, dan CentOS merilis versi baru kira-kira setiap dua tahun.

Saat Anda memulai instance baru di tumpukan CentOS, AWS OpsWorks Stacks secara otomatis menginstal versi CentOS terbaru. Karena AWS OpsWorks Stacks tidak secara otomatis memperbarui sistem operasi pada instance yang ada saat versi minor CentOS baru dirilis, instance yang baru dibuat mungkin menerima versi yang lebih baru daripada instance tumpukan yang ada. Agar versi tetap konsisten di seluruh tumpukan, Anda dapat memperbarui instance yang ada ke versi CentOS saat ini, sebagai berikut:

- Untuk instance online, jalankan perintah tumpukan Sistem Operasi Upgrade, yang berjalan `yum update` pada instance yang ditentukan untuk memperbaruinya ke versi saat ini.

  Ketika versi minor CentOS 7 baru tersedia, halaman Instances dan Stack menampilkan pemberitahuan dengan tautan yang membawa Anda ke halaman Run Command. Anda kemudian dapat menjalankan Upgrade Sistem Operasi untuk meng-upgrade instance Anda.
- Untuk instans offline yang didukung Amazon EBS, mulai instans dan jalankan Sistem Operasi Upgrade seperti yang dijelaskan dalam item daftar sebelumnya.
- Untuk instans yang didukung penyimpanan instans offline, AWS OpsWorks Stacks secara otomatis menginstal versi baru saat instance dimulai ulang.

CentOS: Versi Chef yang Didukung

Versi Koki	Versi CentOS yang Didukung
`12`	`CentOS 7`
`11.10`	`(None supported)`
`11.4 (deprecated)`	`(None supported)`

> **Note**
>
> AWS OpsWorks Stacks mendukung Apache 2.4 untuk instance CentOS.

Linux Red Hat Enterprise

AWS OpsWorks Stacks mendukung versi 64-bit Red Hat Enterprise Linux 7 (RHEL 7). Versi awal yang didukung adalah RHEL 7.1 dan Red Hat merilis versi minor baru kira-kira setiap 9 bulan. Versi minor harus kompatibel dengan RHEL 7.0. Untuk informasi selengkapnya, lihat Kebijakan Siklus Hidup dan Pembaruan.

Saat Anda memulai instance baru, AWS OpsWorks Stacks secara otomatis menginstal versi RHEL 7 saat ini. Karena AWS OpsWorks Stacks tidak secara otomatis memperbarui sistem operasi pada

instance yang ada saat versi minor RHEL 7 baru dirilis, instance yang baru dibuat mungkin menerima versi yang lebih baru daripada instance stack yang ada. Agar versi tetap konsisten di seluruh tumpukan, Anda dapat memperbarui instans yang ada ke versi RHEL 7 saat ini, sebagai berikut:

- Untuk instance online, jalankan perintah tumpukan Sistem Operasi Upgrade, yang berjalan `yum update` pada instance yang ditentukan untuk memperbaruinya ke versi saat ini.

  Ketika versi RHEL 7 baru tersedia, halaman Instances dan Stack menampilkan pemberitahuan dengan tautan yang membawa Anda ke halaman Run Command. Anda kemudian dapat menjalankan Upgrade Sistem Operasi untuk meng-upgrade instance Anda.
- Untuk instans offline yang didukung Amazon EBS, mulai instans dan jalankan Sistem Operasi Upgrade seperti yang dijelaskan dalam item daftar sebelumnya.
- Untuk instans yang didukung penyimpanan instans offline, AWS OpsWorks Stacks secara otomatis menginstal versi baru saat instance dimulai ulang.

Red Hat Enterprise Linux: Versi Node.js yang Didukung

Versi RHEL	Versi Node.js
7	(Node.js versions only apply to Chef 11.10 stacks) 0.8.19 0.8.26 0.10.11 0.10.21 0.10.24 0.10.25 0.10.27 0.10.29 0.10.40 0.12.10 0.12.12 0.12.13 0.12.15

Red Hat Enterprise Linux: Versi Chef yang Didukung

Versi Koki	Versi RHEL yang Didukung
`12`	`Red Hat Enterprise Linux 7`
`11.10`	`Red Hat Enterprise Linux 7`
`11.4 (deprecated)`	`(None supported)`

Semua versi Node.js yang lebih tua dari 0.10.40 tidak digunakan lagi. 0.12.7 dan 0.12.9 juga tidak digunakan lagi.

> **Note**
>
> AWS OpsWorks Stacks mendukung Apache 2.4 untuk instans RHEL 7.

Microsoft Windows Server

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Catatan berikut menjelaskan dukungan AWS OpsWorks Stacks untuk instance Windows. Instans Windows hanya tersedia untuk tumpukan Chef 12.2. Versi yang tepat dari Chef dalam tumpukan Windows adalah 12.22.

Saat ini, agen AWS OpsWorks Stacks tidak dapat diinstal pada—dan AWS OpsWorks Stacks tidak dapat mengelola—instance berbasis Windows yang menggunakan bahasa UI sistem selain Inggris - Amerika Serikat (en-AS).

Versi

AWS OpsWorks Stacks mendukung versi Windows 64-bit berikut:

- Basis Microsoft Windows Server 2022
- Microsoft Windows Server 2022 dengan SQL Server Express
- Microsoft Windows Server 2022 dengan Standar SQL Server
- Microsoft Windows Server 2022 dengan SQL Server Web
- Pangkalan Microsoft Windows Server 2019
- Microsoft Windows Server 2019 dengan SQL Server Express
- Microsoft Windows Server 2019 dengan Standar SQL Server
- Microsoft Windows Server 2019 dengan SQL Server Web

Membuat Instans

Anda membuat instance Windows dengan konsol AWS OpsWorks Stacks, API, atau CLI. Instans Windows didukung Amazon EBS, tetapi Anda tidak dapat memasang volume Amazon EBS tambahan.

Tumpukan Windows dapat menggunakan instance 24/7, yang Anda mulai dan hentikan secara manual. Mereka juga dapat menggunakan penskalaan otomatis berbasis waktu, yang secara otomatis memulai dan menghentikan instance berdasarkan jadwal yang ditentukan pengguna. Tumpukan berbasis Windows tidak dapat menggunakan penskalaan otomatis berbasis beban.

Anda tidak dapat mendaftarkan instance Windows yang dibuat di luar AWS OpsWorks Stacks dengan tumpukan.

Pembaruan

AWS memperbarui Windows AMIs untuk setiap set tambalan, jadi ketika Anda membuat instance, itu akan memiliki pembaruan terbaru. Namun, AWS OpsWorks Stacks tidak menyediakan cara untuk menerapkan pembaruan ke instance Windows online. Cara termudah untuk memastikan bahwa Windows up to date adalah dengan mengganti instans Anda secara teratur, sehingga mereka selalu menjalankan AMI terbaru.

Lapisan

Untuk menangani tugas-tugas seperti menginstal dan mengonfigurasi perangkat lunak atau menyebarkan aplikasi, Anda perlu mengimplementasikan satu atau lebih lapisan khusus dengan resep khusus.

Chef

Instans Windows menggunakan Chef 12.22, dan menjalankan chef-client dalam mode lokal, yang meluncurkan server Chef dalam memori lokal yang disebut chef-zero. Kehadiran server ini memungkinkan resep khusus untuk menggunakan pencarian Chef dan tas data.

Login Jarak Jauh

AWS OpsWorks Stacks menyediakan pengguna IAM resmi dengan kata sandi yang dapat mereka gunakan untuk masuk ke instance Windows. Kata sandi ini kedaluwarsa setelah waktu yang ditentukan. Administrator dapat menggunakan key pair SSH untuk mengambil kata sandi Administrator instance, yang menyediakan akses RDP tanpa batas. Untuk informasi selengkapnya, lihat Masuk dengan RDP.

AWS SDK

AWS OpsWorks Stacks secara otomatis menginstal AWS SDK untuk .NETpada setiap instance. Paket ini mencakup pustaka AWS .NET dan AWS Tools untuk Windows, termasuk AWS Tools for PowerShell. Untuk menggunakan Ruby SDK, Anda dapat meminta resep khusus menginstal permata yang sesuai.

Pemantauan dan Metrik

Instans Windows mendukung metrik Amazon CloudWatch (CloudWatch) standar, yang dapat Anda lihat di konsol. CloudWatch

Ruby

Klien Chef 12.22 yang diinstal AWS OpsWorks Stacks pada instans Windows dilengkapi dengan Ruby 2.3.6. Namun, AWS OpsWorks Stacks tidak menambahkan direktori executable ke variabel lingkungan PATH. Agar aplikasi Anda menggunakan versi Ruby ini, Anda biasanya dapat menemukannya di`C:\opscode\chef\embedded\bin\`.

AWS OpsWorks Agen Tumpukan CLI

Agen AWS OpsWorks Stacks pada instance Windows tidak mengekspos antarmuka baris perintah.

Support Proxy

Lakukan hal berikut untuk menyiapkan dukungan proxy untuk instance Windows:

1. Ubah `machine.config` untuk menambahkan yang berikut ini, yang menambahkan dukungan proxy ke aplikasi Windows PowerShell (bootstrap awal) dan .NET (agen 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. Jalankan perintah berikut untuk mengatur variabel lingkungan untuk digunakan nanti oleh Chef dan 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**
>
> Untuk kontrol lebih lanjut tentang cara AWS OpsWorks Stacks memperbarui instans Anda, buat AMI kustom berdasarkan Microsoft Windows Server 2022 Base. Misalnya, dengan kustom AMIs Anda dapat menentukan perangkat lunak mana yang diinstal pada sebuah instance, seperti Web Server (IIS). Untuk informasi selengkapnya, lihat Menggunakan Custom AMIs.

Menambahkan Instance ke Layer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda membuat layer, Anda biasanya menambahkan setidaknya satu instance. Anda dapat menambahkan lebih banyak instance nanti, jika set saat ini tidak dapat menangani beban. Anda juga dapat menggunakan instans berbasis beban atau berbasis waktu untuk secara otomatis menskalakan jumlah instans.

Anda dapat menambahkan instance baru atau yang sudah ada ke layer:

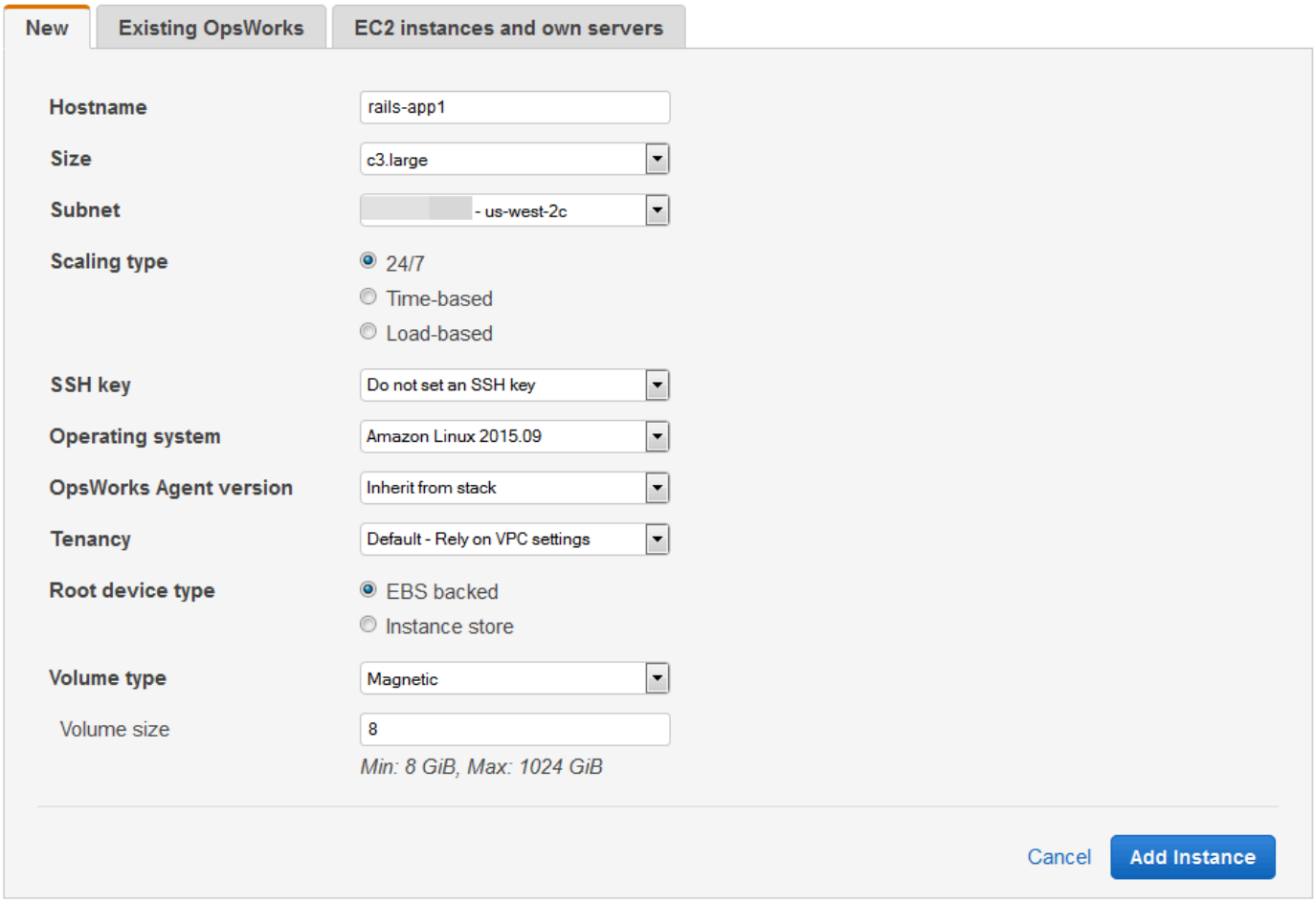
- Baru — OpsWorks membuat instance baru, dikonfigurasi sesuai spesifikasi Anda, dan menjadikannya anggota layer.
- Ada —Anda dapat menambahkan instance yang ada dari lapisan yang kompatibel, tetapi harus dalam keadaan offline (berhenti).

Jika instance milik beberapa lapisan, AWS OpsWorks Stacks menjalankan resep untuk setiap lapisan instance saat peristiwa siklus hidup terjadi, atau saat Anda menjalankan perintah stack atau deployment.

Anda juga dapat membuat instance sebagai anggota dari beberapa lapisan dengan mengedit konfigurasinya. Untuk informasi selengkapnya, lihat Mengedit Konfigurasi Instance.

Untuk menambahkan instance baru ke layer

1. Pada halaman Instances, pilih +Instance untuk layer yang sesuai dan (jika perlu) pilih tab Baru. Jika Anda ingin mengonfigurasi lebih dari sekedar nama Host, Ukuran, dan Subnet atau Availability Zone, pilih Advanced >> untuk melihat opsi lainnya. Berikut ini menunjukkan set lengkap pilihan:



2. Jika diinginkan, Anda dapat mengganti konfigurasi default, yang sebagian besar Anda tentukan saat membuat tumpukan. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

   Nama host

   Mengidentifikasi instance di jaringan. Secara default, AWS OpsWorks Stacks menghasilkan nama host setiap instans dengan menggunakan tema Hostname yang Anda tentukan saat Anda membuat tumpukan. Anda dapat mengganti nilai ini dan menentukan nama host pilihan Anda.

   Size

   Jenis EC2 instans Amazon, yang menentukan sumber daya instans, seperti jumlah memori atau jumlah inti virtual. AWS OpsWorks Stacks menentukan ukuran default untuk setiap instance, yang dapat Anda ganti dengan jenis instans pilihan Anda.

   Jenis instance yang didukung oleh AWS OpsWorks Stacks bergantung pada apakah tumpukan ada di VPC atau tidak. Jenis instans juga terbatas jika akun Anda menggunakan

AWS Tingkat Gratis. Daftar Ukuran drop-down menunjukkan jenis instans yang didukung untuk versi Chef yang didukung stack Anda. Ketahuilah bahwa instance mikro seperti t1.micro mungkin tidak memiliki sumber daya yang cukup untuk mendukung beberapa lapisan. Untuk informasi selengkapnya, lihat Jenis Instance.

> **Note**
>
> Jika Anda menggunakan instans load-balanced, perhatikan bahwa peristiwa siklus hidup Konfigurasi dapat menghasilkan lonjakan beban CPU yang signifikan yang mungkin berlangsung satu menit atau lebih lama. Dengan instance yang lebih kecil, lonjakan beban ini cukup untuk memicu peningkatan, terutama untuk tumpukan besar yang seimbang beban dengan peristiwa Konfigurasi yang sering. Berikut ini adalah beberapa cara untuk mengurangi kemungkinan peristiwa Konfigurasi yang menyebabkan peningkatan yang tidak perlu.
>
> - Gunakan instance yang lebih besar, sehingga beban tambahan dari peristiwa Configure tidak cukup untuk memicu peningkatan skala.
> - Jangan gunakan tipe instance seperti T2 yang berbagi sumber daya CPU.
>
> Ini memastikan bahwa ketika peristiwa Konfigurasi terjadi, semua sumber daya CPU instans segera tersedia.
> - Buat `exceeded threshold` waktu secara signifikan lebih lama dari waktu yang diperlukan untuk memproses acara Konfigurasi, mungkin 5 menit.
>
> Untuk informasi selengkapnya, lihat Menggunakan penskalaan berbasis beban otomatis.

Zona Ketersediaan/Subnet

Jika tumpukan tidak ada dalam VPC, pengaturan ini diberi label Availability Zone dan mencantumkan zona wilayah tersebut. Anda dapat menggunakan pengaturan ini untuk mengganti Availability Zone default yang Anda tentukan saat membuat tumpukan.

Jika tumpukan berjalan di VPC, pengaturan ini diberi label Subnet dan mencantumkan subnet VPC. Anda dapat menggunakan pengaturan ini untuk mengganti subnet default yang Anda tentukan saat membuat tumpukan.

> **Note**
>
> Secara default, AWS OpsWorks Stacks mencantumkan rentang CIDR subnet. Untuk membuat daftar lebih mudah dibaca, gunakan konsol VPC atau API untuk menambahkan tag ke setiap subnet dengan Kunci disetel ke dan Nilai disetel **`Name`** ke nama subnet. AWS OpsWorks Tumpukan menambahkan nama itu ke rentang CIDR. Pada contoh sebelumnya, tag Nama subnet diatur ke. **`Private`**

Jenis Penskalaan

Menentukan bagaimana instance dimulai dan dihentikan.

- Nilai defaultnya adalah instance 24/7, yang Anda mulai dan hentikan secara manual.
- AWS OpsWorks Tumpukan memulai dan menghentikan instance berbasis waktu berdasarkan jadwal yang ditentukan.
- (Hanya Linux) AWS OpsWorks Tumpukan memulai dan menghentikan instance berbasis beban berdasarkan metrik pemuatan yang ditentukan.

> **Note**
>
> Anda tidak memulai atau menghentikan instance berbasis beban atau berbasis waktu sendiri. Sebagai gantinya, Anda mengonfigurasi instance, dan AWS OpsWorks Stacks memulai dan menghentikannya berdasarkan konfigurasi. Untuk informasi selengkapnya, lihat Mengelola beban dengan instans berbasis waktu dan berbasis beban.

Kunci SSH

Sebuah EC2 key pair Amazon. AWS OpsWorks Stacks menginstal kunci publik pada instance.

- Untuk instance Linux, Anda dapat menggunakan kunci pribadi yang sesuai dengan klien SSH untuk masuk ke instance.
- Untuk instance Windows, Anda dapat menggunakan kunci pribadi yang sesuai untuk mengambil kata sandi Administrator instans. Anda kemudian dapat menggunakan kata sandi itu dengan RDP untuk masuk ke instance sebagai Administrator.

Awalnya, pengaturan ini adalah nilai kunci SSH Default yang Anda tentukan saat Anda membuat tumpukan.

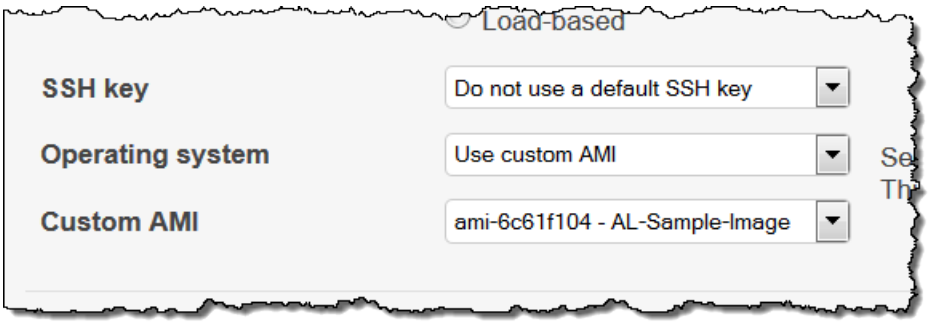
- Jika nilai default diatur ke Jangan gunakan kunci SSH default, Anda dapat menentukan salah satu EC2 kunci Amazon akun Anda.
- Jika nilai default disetel ke EC2 kunci Amazon, Anda dapat menentukan kunci yang berbeda atau tidak ada kunci.

Sistem operasi

Sistem operasi menentukan sistem operasi mana yang dijalankan instance. AWS OpsWorks Stacks hanya mendukung sistem operasi 64-bit.

Awalnya, pengaturan ini adalah nilai sistem operasi Default yang Anda tentukan saat Anda membuat tumpukan. Anda dapat mengganti nilai default untuk menentukan sistem operasi Linux yang berbeda atau Amazon Machine Image (AMI) kustom. Namun, Anda tidak dapat beralih dari Linux ke Windows atau dari Windows ke Linux.

Jika Anda memilih Gunakan AMI kustom, halaman akan menampilkan daftar kustom, AMIs bukan jenis perangkat Arsitektur dan Root.



Untuk informasi selengkapnya, lihat Menggunakan Custom AMIs.

OpsWorks Versi agen

OpsWorks Versi agen menentukan versi agen AWS OpsWorks Stacks yang ingin Anda jalankan pada instance. Jika Anda ingin AWS OpsWorks Stacks memperbarui agen secara otomatis, pilih Inherit dari tumpukan. Untuk menginstal versi agen tertentu, dan memperbarui agen secara manual pada instance, pilih versi dari daftar drop-down.

> **Note**
>
> Tidak semua versi agen bekerja dengan semua rilis sistem operasi. Jika instans Anda menjalankan agen—atau Anda menginstal agen pada instance—yang tidak sepenuhnya didukung pada sistem operasi instans, konsol AWS OpsWorks Stacks menampilkan pesan kesalahan yang menginstruksikan Anda untuk menginstal agen yang kompatibel.

Penghunian

Pilih opsi sewa untuk contoh Anda. Anda dapat memilih untuk menjalankan instance Anda di server fisik yang sepenuhnya didedikasikan untuk Anda gunakan.

- Default - Mengandalkan pengaturan VPC. Tidak ada penyewaan, atau mewarisi pengaturan sewa dari VPC Anda.
- Dedicated - Jalankan instance khusus. Bayar per jam untuk instance yang berjalan pada perangkat keras penyewa tunggal. Untuk informasi selengkapnya, lihat Instans Khusus di Panduan Pengguna Amazon VPC, dan Instans Khusus EC2 Amazon.
- Host khusus - Jalankan instance ini pada host khusus. Bayar host fisik yang sepenuhnya didedikasikan untuk menjalankan instans Anda, dan bawa lisensi perangkat lunak per-soket, per-inti, atau per-VM yang ada untuk mengurangi biaya. Untuk informasi selengkapnya, lihat Ikhtisar Host Khusus di EC2 dokumentasi Amazon, dan Host EC2 Khusus Amazon.

Jenis perangkat root

Menentukan penyimpanan perangkat root instance ini.

- Instans Linux dapat berupa Amazon EBS yang didukung atau didukung oleh toko instans.
- Instans Windows harus didukung Amazon EBS.

Untuk informasi selengkapnya, lihat Penyimpanan.

> **Note**
>
> Setelah boot awal, instans yang didukung Amazon EBS boot lebih cepat daripada instance yang didukung penyimpanan instans karena AWS OpsWorks Stacks

tidak harus menginstal ulang perangkat lunak instans dari awal. Untuk informasi selengkapnya, lihat Penyimpanan Perangkat Root.

Jenis volume

Menentukan jenis volume perangkat root: Magnetic, Provisioned IOPS (SSD), atau General Purpose (SSD). Untuk informasi lebih lanjut, lihat Jenis Volume Amazon EBS..

Ukuran volume

Menentukan ukuran volume perangkat root untuk jenis volume yang ditentukan. Untuk informasi lebih lanjut, lihat Jenis Volume Amazon EBS..

- Tujuan Umum (SSD). Ukuran minimum yang diizinkan adalah: 8 GiB; ukuran maksimum adalah 16384 GiB.
- IOPS (SSD) yang disediakan. Ukuran minimum yang diizinkan adalah: 8 GiB; ukuran maksimum adalah 16384 GiB. Anda dapat mengatur minimal 100 operasi input/output per detik (IOPS), dan maksimum 240 IOPS.
- Magnetik. Ukuran minimum yang diizinkan adalah 8 GiB; ukuran maksimum adalah 1024 GiB.

3. Pilih Add Instance untuk membuat instance baru.

> **Note**
>
> Anda tidak dapat mengganti setelan versi agen default stack saat membuat instance. Untuk menentukan pengaturan versi agen kustom, Anda harus membuat instance dan kemudian mengedit konfigurasinya.

Untuk menambahkan instance yang ada ke lapisan

1. Pada halaman Instances, pilih +Instance untuk layer yang sesuai, lalu buka tab Existing.

> **Note**
>
> Jika Anda berubah pikiran tentang menggunakan instance yang ada, pilih New untuk membuat instance baru seperti yang dijelaskan dalam prosedur sebelumnya.

2. Pada tab yang ada, pilih instance dari daftar.
3. Pilih Add Instance untuk membuat instance baru.

Sebuah instance mewakili EC2 instance Amazon, tetapi pada dasarnya hanya struktur data AWS OpsWorks Stacks. Instance harus dimulai untuk membuat EC2 instance Amazon yang sedang berjalan, seperti yang dijelaskan di bagian berikut.

> **Important**
>
> Jika Anda meluncurkan instance ke VPC default, Anda harus berhati-hati dalam memodifikasi konfigurasi VPC. Instans harus selalu dapat berkomunikasi dengan layanan AWS OpsWorks Stacks, Amazon S3, dan repositori paket. Jika, misalnya, Anda menghapus gateway default, instance akan kehilangan koneksinya ke layanan AWS OpsWorks Stacks, yang kemudian akan memperlakukan instance sebagai gagal dan menyembuhkannya secara otomatis. Namun, AWS OpsWorks Stacks tidak akan dapat menginstal agen instance pada instance yang disembuhkan. Tanpa agen, instans tidak dapat berkomunikasi dengan layanan, dan proses startup tidak akan berkembang melampaui `booting` status. Untuk informasi selengkapnya tentang VPC default, lihat Platform yang Didukung.

Anda juga dapat menggabungkan sumber daya komputasi Linux ke dalam tumpukan yang dibuat di luar AWS OpsWorks Stacks:

- EC2 Instans Amazon yang Anda buat langsung dengan menggunakan EC2 konsol Amazon, CLI, atau API.
- Instans lokal yang berjalan di perangkat keras Anda sendiri, termasuk instance yang berjalan di mesin virtual.

Untuk informasi selengkapnya, lihat Menggunakan Sumber Daya Komputasi yang Dibuat Di Luar AWS OpsWorks Tumpukan.

Menggunakan Custom AMIs

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks mendukung dua cara untuk menyesuaikan instance: Custom Amazon Machine Images (AMIs) dan resep Chef. Kedua pendekatan memberi Anda kontrol atas paket dan versi paket mana yang diinstal, bagaimana mereka dikonfigurasi, dan sebagainya. Namun, setiap pendekatan memiliki keunggulan yang berbeda, jadi yang terbaik tergantung pada kebutuhan Anda.

Berikut ini adalah alasan utama untuk mempertimbangkan menggunakan AMI kustom:

- Anda ingin melakukan pra-paket tertentu alih-alih menginstalnya setelah instance boot.
- Anda ingin mengontrol waktu pembaruan paket untuk memberikan gambar dasar yang konsisten untuk lapisan Anda.
- Anda ingin instance— instance berbasis beban khususnya — untuk boot secepat mungkin.

Berikut ini adalah alasan utama untuk mempertimbangkan menggunakan resep Chef:

- Mereka lebih fleksibel daripada kebiasaan AMIs.
- Mereka lebih mudah diperbarui.
- Mereka dapat melakukan pembaruan pada instance yang sedang berjalan.

Dalam praktiknya, solusi optimal mungkin merupakan kombinasi dari kedua pendekatan. Untuk informasi lebih lanjut tentang resep, lihat Buku Masak dan Resep.

Topik

- Bagaimana Custom AMIs bekerja dengan AWS OpsWorks Stacks
- Membuat AMI Kustom untuk AWS OpsWorks Tumpukan

Bagaimana Custom AMIs bekerja dengan AWS OpsWorks Stacks

Untuk menentukan AMI kustom untuk instans Anda, pilih Gunakan AMI kustom sebagai sistem operasi instans saat Anda membuat instance baru. AWS OpsWorks Tumpukan kemudian menampilkan daftar kustom AMIs di wilayah tumpukan dan Anda memilih yang sesuai dari daftar. Untuk informasi selengkapnya, lihat Menambahkan Instance ke Layer.

> **Note**
>
> Anda tidak dapat menentukan AMI kustom tertentu sebagai sistem operasi default stack. Anda dapat mengatur `Use custom AMI` sebagai sistem operasi default stack, tetapi Anda dapat menentukan AMI tertentu hanya ketika Anda menambahkan instance baru ke lapisan. Untuk informasi selengkapnya, silakan lihat Menambahkan Instance ke Layer dan Membuat Tumpukan Baru. Meskipun dimungkinkan untuk membuat instance dengan sistem operasi lain (seperti CentOS 6. x) yang telah dibuat dari kustom atau yang dihasilkan oleh komunitas AMIs, ini tidak didukung secara resmi.

Topik ini membahas beberapa masalah umum yang harus Anda pertimbangkan sebelum membuat atau menggunakan AMI kustom.

Topik

- Perilaku Startup
- Memilih Layer
- Menangani Aplikasi

Perilaku Startup

Saat memulai instance, AWS OpsWorks Stacks menggunakan AMI kustom yang ditentukan untuk meluncurkan EC2 instans Amazon baru. AWS OpsWorks Stacks kemudian menggunakan cloud-init untuk menginstal agen AWS OpsWorks Stacks pada instance dan agen menjalankan resep Pengaturan instance diikuti oleh resep Deploy. Setelah instance online, agen menjalankan resep Configure untuk setiap instance di stack, termasuk instance yang baru ditambahkan.

Memilih Layer

Agen AWS OpsWorks Stacks biasanya tidak bertentangan dengan paket yang diinstal. Namun, instance harus menjadi anggota setidaknya satu lapisan. AWS OpsWorks Tumpukan selalu menjalankan resep lapisan itu, yang dapat menyebabkan masalah. Anda harus memahami persis apa yang dilakukan resep lapisan pada sebuah instance sebelum menambahkan instance dengan AMI khusus ke lapisan itu.

Untuk melihat resep mana yang dijalankan oleh tipe layer tertentu pada instance Anda, buka tumpukan yang menyertakan lapisan itu. Lalu klik Layers di panel navigasi, dan klik Resep untuk layer yang diminati. Untuk melihat kode yang sebenarnya, klik nama resep.

> **Note**
>
> Untuk Linux AMIs, salah satu cara untuk mengurangi kemungkinan konflik adalah dengan menggunakan AWS OpsWorks Stacks untuk menyediakan dan mengkonfigurasi instance yang menjadi dasar untuk AMI kustom Anda. Untuk informasi selengkapnya, lihat Buat AMI Linux Kustom dari Instans AWS OpsWorks Stacks.

Menangani Aplikasi

Selain paket, Anda mungkin juga ingin menyertakan aplikasi di AMI. Jika Anda memiliki aplikasi kompleks yang besar, memasukkannya ke dalam AMI dapat mempersingkat waktu startup instans. Anda dapat menyertakan aplikasi kecil di AMI Anda, tetapi biasanya ada sedikit atau tidak ada keuntungan waktu dibandingkan dengan memiliki AWS OpsWorks Stacks menyebarkan aplikasi.

Salah satu opsi adalah memasukkan aplikasi ke dalam AMI Anda dan juga membuat aplikasi yang menyebarkan aplikasi ke instance dari repositori. Pendekatan ini mempersingkat waktu boot Anda tetapi juga menyediakan cara mudah untuk memperbarui aplikasi setelah instance berjalan. Perhatikan bahwa resep Chef adalah idempoten, sehingga resep penerapan tidak akan memodifikasi aplikasi selama versi dalam repositori sama dengan yang ada di instance.

Membuat AMI Kustom untuk AWS OpsWorks Tumpukan

Untuk menggunakan AMI kustom dengan AWS OpsWorks Stacks, Anda harus terlebih dahulu membuat AMI dari instance yang disesuaikan. Anda dapat memilih dari dua opsi:

- Gunakan EC2 konsol Amazon atau API untuk membuat dan menyesuaikan instance, berdasarkan versi 64-bit dari salah satu AWS OpsWorks Stacks yang didukung AMIs.
- Untuk Linux AMIs, gunakan OpsWorks untuk membuat EC2 instance Amazon, berdasarkan konfigurasi lapisan terkait.

Sebelum Anda membuat AMI Linux kustom, nonaktifkan `noexec` pada `/tmp` partisi untuk memungkinkan AWS OpsWorks Stacks menginstal agennya pada instance Linux kustom.

> **Note**
>
> Ketahuilah bahwa AMI mungkin tidak berfungsi dengan semua tipe instans, jadi pastikan AMI awal Anda kompatibel dengan tipe instans yang akan Anda gunakan. Secara khusus, tipe instans R3 memerlukan AMI virtualisasi berbantuan perangkat keras (HVM).

Anda kemudian menggunakan EC2 konsol Amazon atau API untuk membuat AMI kustom dari instance yang disesuaikan. Anda dapat menggunakan kustom Anda AMIs di tumpukan apa pun yang berada di wilayah yang sama dengan menambahkan instance ke lapisan dan menentukan AMI kustom Anda. Untuk informasi selengkapnya tentang cara membuat instance yang menggunakan AMI kustom, lihatMenambahkan Instance ke Layer.

> **Note**
>
> Secara default, AWS OpsWorks Stacks menginstal semua pembaruan Amazon Linux saat boot, yang memberi Anda rilis terbaru. Selain itu, Amazon Linux merilis versi baru kira-kira setiap enam bulan, yang dapat melibatkan perubahan signifikan. Secara default, kustom AMIs berbasis Amazon Linux secara otomatis diperbarui ke versi baru ketika dirilis. Praktik yang disarankan adalah mengunci AMI kustom Anda ke versi Amazon Linux tertentu, yang memungkinkan Anda menunda pembaruan hingga Anda menguji versi baru. Untuk informasi selengkapnya, lihat Bagaimana cara mengunci AMI ke versi tertentu? .

Topik

- Buat AMI Kustom menggunakan Amazon EC2
- Buat AMI Linux Kustom dari Instans AWS OpsWorks Stacks
- Buat AMI Windows Kustom

Buat AMI Kustom menggunakan Amazon EC2

Cara paling sederhana untuk membuat AMI kustom—dan satu-satunya pilihan untuk AMIs Windows—adalah melakukan seluruh tugas dengan menggunakan EC2 konsol Amazon atau API. Untuk detail selengkapnya tentang langkah-langkah berikut, lihat Membuat Sendiri AMIs.

Untuk membuat AMI kustom menggunakan EC2 konsol Amazon atau API

1. Buat instance dengan menggunakan versi 64-bit dari salah satu AWS OpsWorks Stacks yang didukung AMIs.
2. Sesuaikan instance dari Langkah 1 dengan mengonfigurasinya, menginstal paket, dan sebagainya. Ingat bahwa semua yang Anda instal akan direproduksi pada setiap instance berdasarkan AMI, jadi jangan sertakan item yang harus spesifik untuk instance tertentu.
3. Hentikan instance dan buat AMI kustom.

Buat AMI Linux Kustom dari Instans AWS OpsWorks Stacks

Untuk menggunakan instance AWS OpsWorks Stacks Linux yang disesuaikan untuk membuat AMI, ketahuilah bahwa setiap EC2 instans Amazon yang dibuat oleh OpsWorks menyertakan identitas unik. Jika Anda membuat AMI kustom dari instance semacam itu, itu menyertakan identitas itu, dan semua instance berdasarkan AMI memiliki identitas yang sama. Untuk memastikan bahwa instans berdasarkan AMI kustom Anda memiliki identitas unik, Anda harus menghapus identitas dari instance yang disesuaikan sebelum membuat AMI.

Untuk membuat AMI kustom dari instance AWS OpsWorks Stacks

1. Buat tumpukan Linux dan tambahkan satu atau lebih lapisan untuk menentukan konfigurasi instance yang disesuaikan. Anda dapat menggunakan lapisan bawaan, disesuaikan sesuai kebutuhan, serta lapisan yang sepenuhnya disesuaikan. Untuk informasi selengkapnya, lihat Menyesuaikan Tumpukan AWS OpsWorks.
2. Edit lapisan dan nonaktifkan AutoHealing.
3. Tambahkan instance dengan distribusi Linux pilihan Anda ke layer atau layer dan mulailah. Sebaiknya gunakan instans yang didukung Amazon EBS. Buka halaman detail instans dan catat EC2 ID Amazon untuk nanti.
4. Saat instans online, masuk dengan SSH, dan lakukan salah satu dari empat langkah berikutnya, tergantung pada sistem operasi instans Anda.
5. Untuk instance Amazon Linux di tumpukan Chef 11 atau Chef 12, atau instance Red Hat Enterprise Linux 7 di tumpukan Chef 11, lakukan hal berikut.

   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**
>
> Untuk contoh di tumpukan Chef 12, tambahkan dua folder berikut ke perintah ini:
>
> - `/var/chef`
> - `/opt/chef`

d. `sudo rpm -e opsworks-agent-ruby`

e. `sudo rpm -e chef`

6. Untuk instance Ubuntu 16.04 LTS atau 18.04 LTS dalam tumpukan Chef 12, lakukan hal berikut.

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. Untuk versi Ubuntu lain yang didukung dalam tumpukan Chef 12, lakukan hal berikut.

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. Untuk contoh Red Hat Enterprise Linux 7 di tumpukan Chef 12, lakukan hal berikut.

   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. Langkah ini tergantung pada jenis instance:

   - Untuk instans yang didukung Amazon EBS, gunakan konsol AWS OpsWorks Stacks untuk menghentikan instans dan membuat AMI seperti yang dijelaskan dalam Membuat AMI Linux yang didukung Amazon EBS..
   - Untuk instance yang didukung toko, buat AMI seperti yang dijelaskan dalam Membuat AMI Linux yang Didukung Toko Instance dan kemudian gunakan konsol Stacks AWS OpsWorks untuk menghentikan instance.

     Saat Anda membuat AMI, pastikan untuk menyertakan file sertifikat. Misalnya, Anda dapat memanggil `ec2-bundle-vol`perintah dengan `-i` argumen yang disetel ke`-i $(find /etc /usr /opt -name '*.pem' -o -name '*.crt' -o -name '*.gpg' | tr '\n' ',')`. Jangan hapus kunci publik apt saat bundling. `ec2-bundle-vol`Perintah default menangani tugas ini.

10. Bersihkan tumpukan Anda dengan kembali ke konsol AWS OpsWorks Stacks dan menghapus instance dari tumpukan.

Buat AMI Windows Kustom

Prosedur berikut membuat kustom AMIs untuk Windows Server 2022 Base. Anda dapat memilih sistem operasi Windows Server lainnya di konsol EC2 manajemen Amazon.

> **Important**
>
> Saat ini, agen AWS OpsWorks Stacks tidak dapat diinstal pada—dan AWS OpsWorks Stacks tidak dapat mengelola—instance berbasis Windows yang menggunakan bahasa UI sistem selain Inggris - Amerika Serikat (en-AS).

Topik

- Membuat AMI Windows Kustom dengan Sysprep
- Membuat AMI Windows Kustom Tanpa Sysprep
- Menambahkan Instans Baru dengan Menggunakan AMI Windows Kustom

Membuat AMI Windows Kustom dengan **`Sysprep`**

Membuat Windows kustom AMIs dengan menggunakan Sysprep biasanya menghasilkan peluncuran instance yang lebih lambat, tetapi proses yang lebih bersih. Startup pertama kali instance yang dibuat dari gambar yang dibuat `Sysprep` membutuhkan lebih banyak waktu karena `Sysprep` aktivitas, restart, penyediaan Stacks, dan AWS OpsWorks Stacks pertama yang dijalankan, termasuk AWS OpsWorks penyiapan dan konfigurasi. Selesaikan langkah-langkah untuk membuat AMI Windows khusus di EC2 konsol Amazon.

Untuk membuat AMI Windows kustom dengan Sysprep

1. Di EC2 konsol Amazon, pilih Launch Instance.
2. Temukan Microsoft Windows Server 2022 Base, lalu pilih Pilih.
3. Pilih jenis instance yang Anda inginkan, lalu pilih Configure Instance Details. Buat perubahan konfigurasi pada AMI, termasuk nama mesin, penyimpanan, dan pengaturan grup keamanan. Pilih Luncurkan.
4. Setelah proses boot instance selesai, dapatkan kata sandi Anda, lalu sambungkan ke instance di jendela Windows Remote Desktop Connection.
5. Pada layar Mulai Windows, pilih Mulai, dan kemudian mulai mengetik **`ec2configservice`** hingga hasilnya menunjukkan EC2ConfigServiceSettingskonsol. Buka konsol .
6. Pada tab Umum, pastikan kotak centang Aktifkan UserData eksekusi terisi (meskipun opsi ini tidak diperlukan`Sysprep`, AWS OpsWorks Stacks harus menginstal agennya). Kosongkan kotak centang untuk Mengatur nama komputer instance... pilihan, karena opsi ini dapat menyebabkan restart loop dengan AWS OpsWorks Stacks.

7. Pada tab Gambar, atur Kata Sandi Administrator ke Acak untuk memungkinkan Amazon EC2 secara otomatis menghasilkan kata sandi yang dapat Anda ambil dengan kunci SSH, atau Tentukan untuk menentukan kata sandi Anda sendiri. `Sysprep`menyimpan pengaturan ini. Jika Anda menentukan kata sandi Anda sendiri, simpan kata sandi di tempat yang nyaman. Kami menyarankan Anda untuk tidak memilih Keep Existing.
8. Pilih Terapkan, lalu pilih Shutdown dengan Sysprep. Ketika Anda diminta untuk mengonfirmasi, pilih Ya.
9. Setelah instance berhenti, di EC2 konsol Amazon, klik kanan instance di daftar Instans, pilih Gambar, lalu pilih Buat Gambar.
10. Pada halaman Buat Gambar, berikan nama dan deskripsi untuk gambar, dan tentukan konfigurasi volume. Setelah selesai, pilih Buat Gambar.
11. Buka halaman Gambar, dan tunggu gambar Anda berubah dari tahap tertunda menjadi tersedia. AMI baru Anda siap digunakan.

Membuat AMI Windows Kustom Tanpa **`Sysprep`**

Selesaikan langkah-langkah untuk membuat AMI Windows khusus di EC2 konsol Amazon.

Untuk membuat AMI Windows kustom tanpa Sysprep

1. Di EC2 konsol Amazon, pilih Launch Instance.
2. Temukan Microsoft Windows Server 2022 Base, lalu pilih Pilih.
3. Pilih jenis instance yang Anda inginkan, lalu pilih Configure Instance Details. Buat perubahan konfigurasi pada AMI, termasuk nama mesin, penyimpanan, dan pengaturan grup keamanan. Pilih Luncurkan.
4. Setelah proses boot instance selesai, dapatkan kata sandi Anda, lalu sambungkan ke instance di jendela Windows Remote Desktop Connection.
5. Pada contoh, buka`C:\Program Files\Amazon\Ec2ConfigService\Settings \config.xml`, ubah dua pengaturan berikut, lalu simpan dan tutup file:
   - `Ec2SetPassword` untuk `Enabled`
   - `Ec2HandleUserData` untuk `Enabled`
6. Putuskan sambungan dari sesi Remote Desktop, dan kembali ke EC2 konsol Amazon.
7. Dalam daftar Instances, hentikan instance.

8. Setelah instance berhenti, di EC2 konsol Amazon, klik kanan instance di daftar Instans, pilih Gambar, lalu pilih Buat Gambar.
9. Pada halaman Buat Gambar, berikan nama dan deskripsi untuk gambar, dan tentukan konfigurasi volume. Setelah selesai, pilih Buat Gambar.
10. Buka halaman Gambar, dan tunggu gambar Anda berubah dari tahap tertunda menjadi tersedia. AMI baru Anda siap digunakan.

Menambahkan Instans Baru dengan Menggunakan AMI Windows Kustom

Setelah gambar Anda berubah ke status yang tersedia, Anda dapat membuat instance baru yang didasarkan pada AMI Windows kustom Anda. Bila Anda memilih Use custom Windows AMI dari daftar sistem operasi, AWS OpsWorks Stacks menampilkan daftar kustom AMIs.

Untuk menambahkan instance baru berdasarkan AMI Windows kustom

1. Saat AMI baru Anda tersedia, buka konsol AWS OpsWorks Stacks, buka halaman Instances untuk tumpukan Windows, dan pilih + Instance di dekat bagian bawah halaman untuk menambahkan instance baru.
2. Pada tab Baru, pilih Advanced.
3. Pada daftar drop-down Sistem operasi, pilih Use custom Windows AMI.
4. Pada daftar drop-down AMI Kustom, pilih AMI yang Anda buat, lalu pilih Tambah Instance.

Anda sekarang dapat memulai dan menjalankan instance.

Memulai, Menghentikan, dan Memulai Ulang Instans 24/7 Secara Manual

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Anda dapat menggunakan instance 24/7 dengan tumpukan Linux dan Windows.

Setelah menambahkan instance 24/7 ke layer, Anda harus memulai instans secara manual untuk meluncurkan instans Amazon Elastic Compute Cloud EC2 (Amazon) yang sesuai dan menghentikannya secara manual untuk menghentikan instans Amazon. EC2 Anda juga dapat me-reboot instance secara manual yang tidak berfungsi dengan baik. AWS OpsWorks Tumpukan secara otomatis memulai dan menghentikan instance berbasis waktu dan berbasis beban. Untuk informasi selengkapnya, lihat Mengelola beban dengan instans berbasis waktu dan berbasis beban.

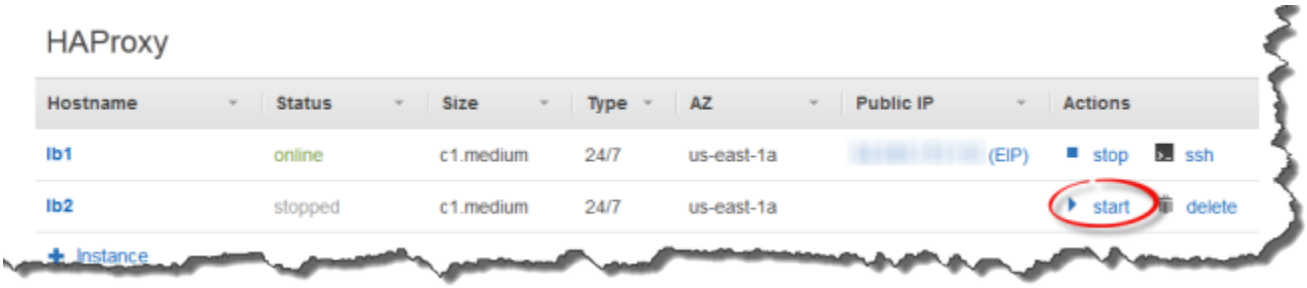
> **Important**
>
> AWS OpsWorks Instans tumpukan harus dimulai, dihentikan, dan dimulai ulang hanya di konsol. AWS OpsWorks AWS OpsWorks tidak mengenali operasi mulai, berhenti, atau memulai ulang yang dilakukan di EC2 konsol Amazon.

Topik

- Memulai atau Memulai Ulang Instance
- Menghentikan sebuah Instance
- Mem-boot Ulang Instans

Memulai atau Memulai Ulang Instance

Untuk memulai instance baru, pada halaman Instans, klik mulai di kolom Tindakan instans.



Anda juga dapat membuat beberapa instance dan kemudian memulai semuanya secara bersamaan dengan mengklik Mulai semua Instans.

Setelah Anda memulai instance, AWS OpsWorks Stacks meluncurkan EC2 instans Amazon dan mem-boot sistem operasi. Proses startup biasanya memakan waktu beberapa menit, dan biasanya agak lebih lambat untuk instance Windows daripada untuk instance Linux. Saat startup berlangsung, bidang Status instance menampilkan rangkaian nilai berikut:

1. request - AWS OpsWorks Stacks telah memanggil EC2 layanan Amazon untuk membuat EC2 instance Amazon.
2. pending - AWS OpsWorks Stacks sedang menunggu EC2 instans Amazon dimulai.
3. booting - EC2 Instans Amazon sedang booting.
4. running_setup - AWS OpsWorks Stacks telah memicu peristiwa Setup dan menjalankan layer Setup resep, diikuti oleh Deploy resep. Untuk informasi selengkapnya, lihat Resep Eksekusi. Jika Anda telah menambahkan buku masak khusus ke tumpukan, AWS OpsWorks Stacks menginstal versi saat ini dari repositori Anda sebelum menjalankan Setup and Deploy resep.
5. online - Instance siap digunakan.

Ketika Status berubah menjadi online, instans beroperasi penuh.

- Jika layer memiliki load balancer terpasang, AWS OpsWorks Stacks menambahkan instance ke dalamnya.
- AWS OpsWorks Tumpukan memicu a Configure event, yang menjalankan setiap instance Configure resep.

  Sesuai kebutuhan, resep ini memperbarui instance untuk mengakomodasi instance baru.
- AWS OpsWorks Stacks menggantikan aksi start instance dengan stop, yang dapat Anda gunakan untuk menghentikan instance.

Jika instance tidak berhasil dimulai atau resep penyiapan gagal, status masing-masing akan disetel ke start_failed atau setup_failed. Anda dapat memeriksa log untuk menentukan penyebabnya. Untuk informasi selengkapnya, lihat Panduan Debugging dan Pemecahan Masalah.

Instance yang dihentikan tetap menjadi bagian dari tumpukan dan mempertahankan semua sumber daya. Misalnya, volume Amazon EBS dan alamat IP Elastis masih dikaitkan dengan instans yang dihentikan. Anda dapat memulai ulang instance yang dihentikan dengan memilih start di kolom Actions instance. Memulai ulang instance yang dihentikan melakukan hal berikut:

- Instans yang didukung toko instans — AWS OpsWorks Stacks meluncurkan instans EC2 Amazon baru dengan konfigurasi yang sama.
- Instans yang didukung Amazon EBS — AWS OpsWorks Stacks memulai ulang EC2 instans Amazon, yang memasang kembali volume root.

Setelah instance selesai booting, AWS OpsWorks Stacks menginstal pembaruan sistem operasi dan menjalankan Setup and Deploy resep, sama seperti awal awal. AWS OpsWorks Stacks juga melakukan hal berikut untuk instance yang dimulai ulang, sebagaimana mestinya.

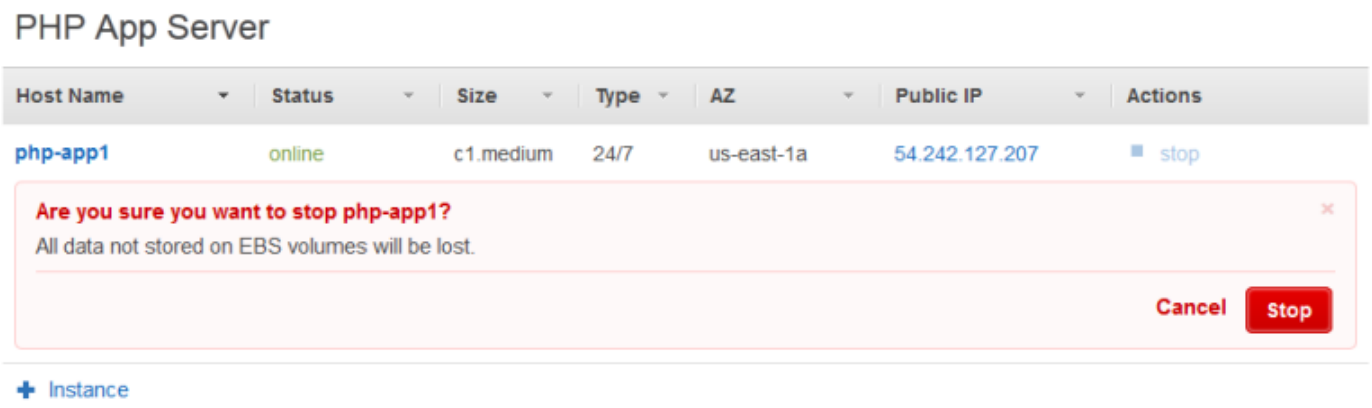
- Mengasosiasikan kembali alamat IP Elastis.
- Memasang kembali volume Amazon Elastic Block Store (Amazon EBS).
- Misalnya instance yang didukung toko, menginstal versi buku masak terbaru.

  Instans yang didukung Amazon EBS terus menggunakan buku masak khusus yang disimpan pada volume root. Jika buku masak kustom Anda telah berubah sejak Anda menghentikan instance, Anda harus memperbaruinya secara manual setelah instance online. Untuk informasi selengkapnya, lihat Memperbarui Buku Masak Kustom.

> **Note**
>
> Mungkin diperlukan beberapa menit untuk alamat IP Elastis untuk diasosiasikan kembali dengan instance yang dimulai ulang. Ketahuilah bahwa pengaturan Elastic IP instance mewakili metadata, dan hanya menunjukkan bahwa alamat harus dikaitkan dengan instance. Pengaturan IP Publik mencerminkan status instans, dan mungkin kosong pada awalnya. Ketika alamat IP Elastis dikaitkan dengan instance, alamat ditetapkan ke pengaturan IP Publik, diikuti oleh (EIP).

Menghentikan sebuah Instance

Pada halaman Instances, klik stop di kolom Actions instance, yang memberi tahu AWS OpsWorks Stacks untuk menjalankan resep shutdown dan menghentikan instance. EC2



Anda juga dapat mematikan setiap instance di tumpukan dengan mengklik Stop All Instances.

Setelah Anda menghentikan instance, AWS OpsWorks Stacks melakukan beberapa tugas:

1. Jika layer instance memiliki penyeimbang beban Elastic Load Balancing yang terpasang, AWS OpsWorks Stacks membatalkan pendaftaran instance.

   Jika layer mendukung fitur pengurasan koneksi penyeimbang beban, AWS OpsWorks Stacks menunda memicu Shutdown acara sampai pengeringan koneksi selesai. Untuk informasi selengkapnya, lihat Lapisan Elastic Load Balancing.
2. AWS OpsWorks Tumpukan memicu a Shutdown event, yang menjalankan instance Shutdown resep.
3. Setelah memicu Shutdown event, AWS OpsWorks Stacks menunggu waktu tertentu untuk memungkinkan Shutdown resep waktu untuk menyelesaikan dan kemudian melakukan hal berikut:
   - Mengakhiri instance yang didukung penyimpanan instance, yang menghapus semua data.
   - Menghentikan instans yang didukung Amazon EBS, yang menyimpan data pada volume root.

   Untuk informasi selengkapnya tentang penyimpanan instans, lihat Penyimpanan.

> **Note**
>
> Pengaturan batas waktu shutdown default adalah 120 detik. Jika Shutdown resep membutuhkan lebih banyak waktu, Anda dapat mengedit konfigurasi lapisan untuk mengubah pengaturan.

Anda dapat memantau proses shutdown dengan melihat kolom Status instans. Saat shutdown berlangsung, ini menampilkan serangkaian nilai berikut:

1. terminating - AWS OpsWorks Stacks menghentikan instance Amazon. EC2
2. shutting_down - AWS OpsWorks Stacks menjalankan layer Shutdown resep.
3. dihentikan - EC2 Instans Amazon dihentikan.
4. berhenti - Instance telah berhenti.

Mem-boot Ulang Instans

Pada halaman Instans, klik nama instans yang tidak berfungsi untuk membuka halaman detail dan kemudian klik Reboot.



Perintah ini melakukan soft reboot dari EC2 instance Amazon terkait. Itu tidak menghapus data instance, bahkan misalnya instance yang didukung penyimpanan, dan tidak memicu peristiwa siklus hidup apa pun.

> **Note**
>
> Agar AWS OpsWorks Stacks secara otomatis mengganti instance yang gagal, aktifkan penyembuhan otomatis. Untuk informasi selengkapnya, lihat Menggunakan Penyembuhan Otomatis.

Mengelola beban dengan instans berbasis waktu dan berbasis beban

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Karena lalu lintas masuk Anda bervariasi, tumpukan Anda mungkin memiliki terlalu sedikit instance untuk menangani beban dengan nyaman atau lebih banyak instance daripada yang diperlukan. Anda dapat menghemat waktu dan uang dengan menggunakan instance berbasis waktu atau berbasis beban untuk secara otomatis menambah atau mengurangi instance lapisan sehingga Anda selalu memiliki cukup instance untuk menangani lalu lintas masuk secara memadai tanpa membayar kapasitas yang tidak dibutuhkan. Tidak perlu memantau beban server atau memulai atau menghentikan instance secara manual. Selain itu, instans berbasis waktu dan beban secara otomatis mendistribusikan, menskalakan, dan menyeimbangkan aplikasi melalui beberapa Availability Zone dalam suatu wilayah, memberi Anda redundansi geografis dan skalabilitas.

Penskalaan otomatis didasarkan pada dua jenis instance, yang menyesuaikan instance online layer berdasarkan kriteria yang berbeda:

- Contoh berbasis waktu

  Mereka memungkinkan tumpukan untuk menangani beban yang mengikuti pola yang dapat diprediksi dengan memasukkan instance yang berjalan hanya pada waktu tertentu atau pada hari-hari tertentu. Misalnya, Anda dapat memulai beberapa contoh setelah jam 6 sore untuk melakukan tugas pencadangan setiap malam atau menghentikan beberapa contoh pada akhir pekan ketika lalu lintas lebih rendah.
- Instans berbasis beban

  Mereka memungkinkan tumpukan untuk menangani beban variabel dengan memulai contoh tambahan ketika lalu lintas tinggi dan menghentikan instance ketika lalu lintas rendah, berdasarkan salah satu dari beberapa metrik beban. Misalnya, Anda dapat memiliki instance AWS OpsWorks Stacks start ketika penggunaan CPU rata-rata melebihi 80% dan menghentikan instance ketika beban CPU rata-rata turun di bawah 60%.

Instans berbasis waktu dan berbasis beban didukung untuk tumpukan Linux, sementara hanya instance berbasis waktu yang didukung untuk tumpukan Windows.

Tidak seperti contoh 24/7, yang harus Anda mulai dan hentikan secara manual, Anda tidak memulai atau menghentikan instance berbasis waktu atau berbasis beban sendiri. Sebagai gantinya, Anda mengonfigurasi instance dan AWS OpsWorks Stacks memulai atau menghentikannya berdasarkan

konfigurasinya. Misalnya, Anda mengonfigurasi instance berbasis waktu untuk memulai dan berhenti pada jadwal yang ditentukan. AWS OpsWorks Tumpukan kemudian memulai dan menghentikan instance sesuai dengan konfigurasi itu.

Praktik yang umum adalah menggunakan ketiga jenis instance bersama-sama, sebagai berikut.

- Instans 24/7 yang ditetapkan untuk menangani beban dasar. Anda biasanya hanya memulai contoh ini dan membiarkannya berjalan terus menerus.
- Satu set instance berbasis waktu, yang AWS OpsWorks Stacks dimulai dan berhenti untuk menangani variasi lalu lintas yang dapat diprediksi. Misalnya, jika lalu lintas Anda tertinggi selama jam kerja, Anda akan mengonfigurasi instance berbasis waktu untuk memulai di pagi hari dan ditutup di malam hari.
- Satu set instance berbasis beban, yang AWS OpsWorks Stacks dimulai dan dihentikan untuk menangani variasi lalu lintas yang tidak terduga. AWS OpsWorks Tumpukan memulainya ketika beban mendekati kapasitas tumpukan 24/7 dan instance berbasis waktu, dan menghentikannya ketika lalu lintas kembali normal.

Untuk informasi selengkapnya tentang cara menggunakan waktu penskalaan ini, lihatMengoptimalkan Jumlah Server.

> **Note**
>
> Jika Anda telah membuat aplikasi untuk lapisan instans atau membuat buku masak khusus, AWS OpsWorks Stacks secara otomatis menerapkan versi terbaru ke instance berbasis waktu dan berbasis beban saat pertama kali dimulai. Namun, AWS OpsWorks Stacks tidak serta merta menyebarkan buku masak terbaru untuk memulai ulang instance offline. Untuk informasi selengkapnya, silakan lihat Mengedit Aplikasi dan Memperbarui Buku Masak Kustom.

Topik

- Menggunakan penskalaan berbasis waktu otomatis
- Menggunakan penskalaan berbasis beban otomatis
- Bagaimana penskalaan berbasis beban berbeda dari penyembuhan otomatis

Menggunakan penskalaan berbasis waktu otomatis

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Penskalaan berbasis waktu memungkinkan Anda mengontrol berapa banyak instance yang harus dimiliki lapisan online pada waktu-waktu tertentu dalam sehari atau hari dalam seminggu dengan memulai atau menghentikan instance pada jadwal yang ditentukan. AWS OpsWorks Stacks memeriksa setiap beberapa menit dan memulai atau menghentikan instance sesuai kebutuhan. Anda menentukan jadwal secara terpisah untuk setiap contoh, sebagai berikut:

- Waktu hari. Anda dapat memiliki lebih banyak instance yang berjalan di siang hari daripada di malam hari, misalnya.
- Hari dalam seminggu. Anda dapat memiliki lebih banyak instance yang berjalan pada hari kerja daripada akhir pekan, misalnya.

> **Note**
>
> Anda tidak dapat menentukan tanggal tertentu.

Topik

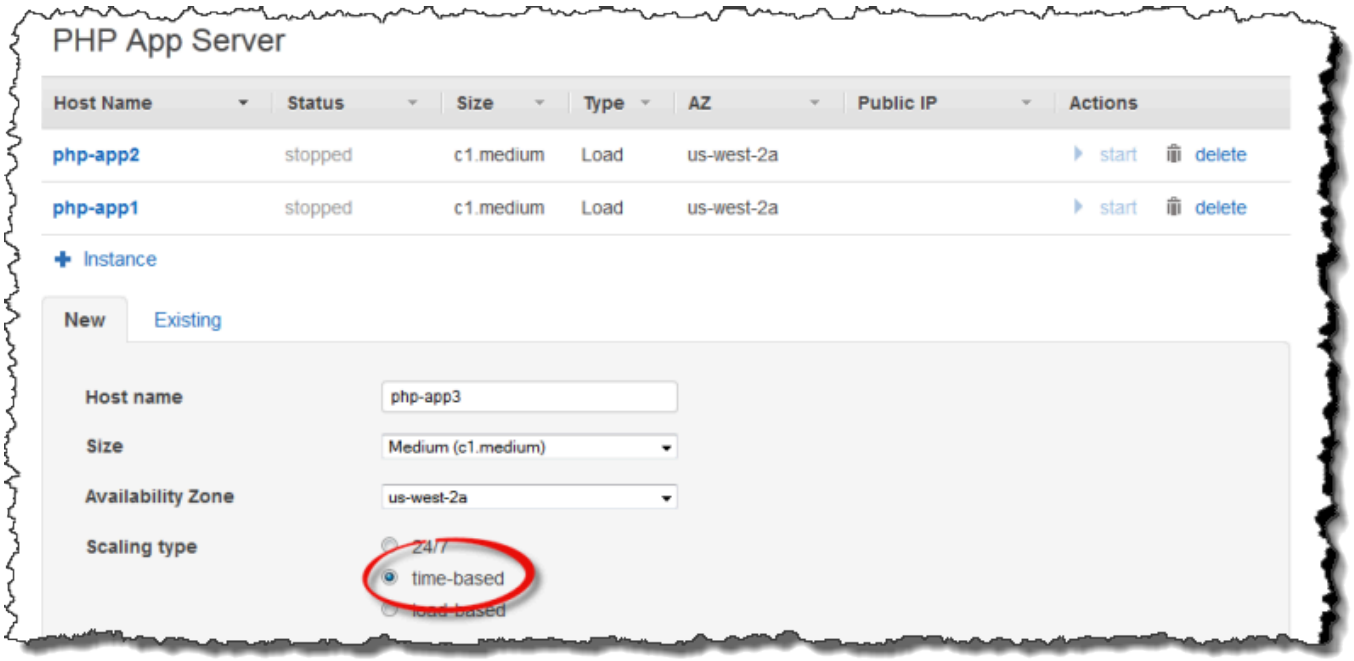
- Menambahkan instance berbasis waktu ke lapisan
- Mengkonfigurasi instance berbasis waktu

Menambahkan instance berbasis waktu ke lapisan

Anda dapat menambahkan instance berbasis waktu baru ke lapisan, atau menggunakan instance yang sudah ada.

Untuk menambahkan instance berbasis waktu baru

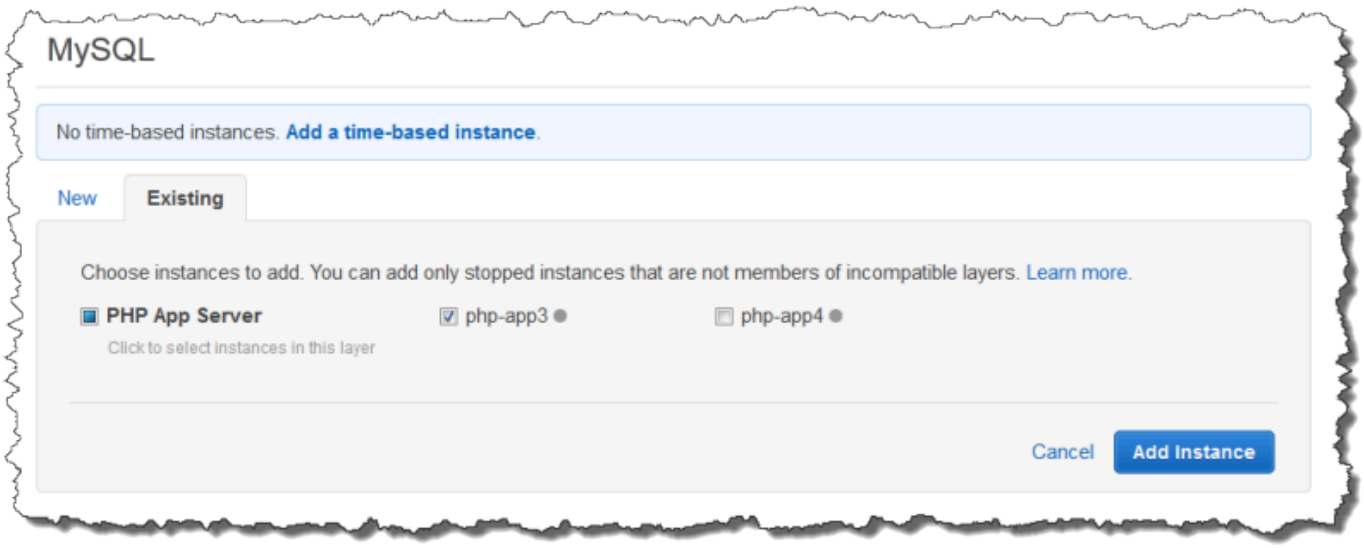
1. Pada halaman Instances, pilih + Instance untuk menambahkan instance. Pada tab Baru, pilih Advanced, lalu pilih berbasis waktu.



2. Konfigurasikan instance. Kemudian pilih Add Instance untuk menambahkan instance ke layer.

Untuk menambahkan instance berbasis waktu yang ada ke lapisan

1. Pada halaman Instans Berbasis Waktu, pilih + Instance jika layer sudah memiliki instance berbasis waktu. Jika tidak, pilih Tambahkan instance berbasis waktu. Kemudian pilih tab yang ada.



2. Pada tab Existing, pilih instance dari daftar. Daftar ini hanya menampilkan instance berbasis waktu.

> **Note**
>
> Jika Anda berubah pikiran tentang menggunakan instance yang ada, pada tab Baru, buat instance baru, seperti yang dijelaskan dalam prosedur sebelumnya.

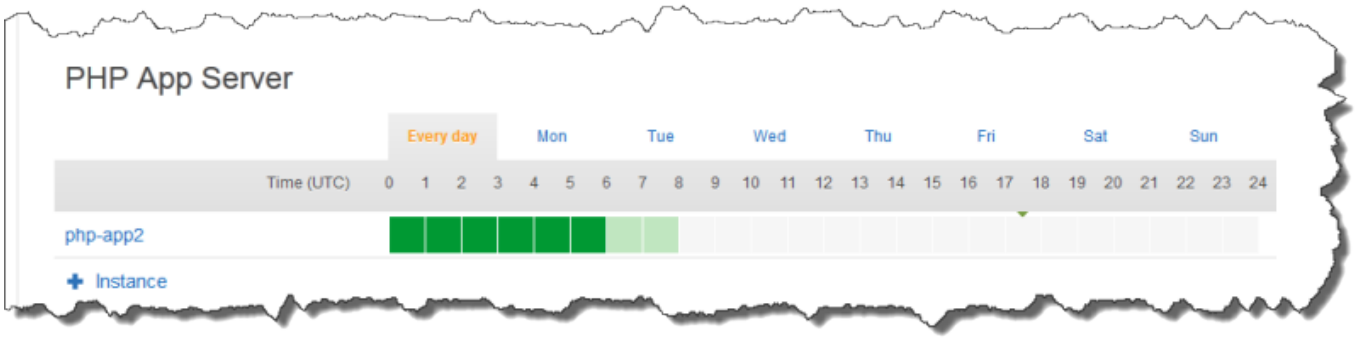
3. Pilih Add instance untuk menambahkan instance ke layer.

Mengkonfigurasi instance berbasis waktu

Setelah Anda menambahkan instance berbasis waktu ke lapisan, Anda mengonfigurasi jadwalnya sebagai berikut.

Untuk mengkonfigurasi instance berbasis waktu

1. Di panel navigasi, di bawah Instans, pilih Berbasis waktu.
2. Tentukan periode online untuk setiap instance berbasis waktu dengan mengisi kotak yang sesuai di bawah jam yang diinginkan.
   - Untuk menggunakan jadwal yang sama setiap hari, pilih tab Setiap hari, lalu tentukan periode waktu online.
   - Untuk menggunakan jadwal yang berbeda pada hari yang berbeda, pilih setiap hari, dan kemudian pilih periode waktu yang sesuai.



> **Note**
>
> Pastikan untuk mengizinkan jumlah waktu yang diperlukan untuk memulai sebuah instance, dan AWS OpsWorks Stacks hanya memeriksa setiap beberapa menit untuk melihat apakah instance harus dimulai atau dihentikan. Misalnya, jika sebuah instance harus berjalan pada 1:00 UTC, mulailah pada pukul 0:00 UTC. Jika tidak, AWS OpsWorks Stacks mungkin tidak memulai instance sampai beberapa menit melewati 1:00 UTC, dan instance membutuhkan beberapa menit lagi untuk online.

Anda dapat mengubah periode waktu online instans kapan saja dengan melakukan langkah-langkah sebelumnya. Saat berikutnya AWS OpsWorks Stacks memeriksa, ia menggunakan jadwal baru untuk menentukan apakah akan memulai atau menghentikan instance.

> **Note**
>
> Anda dapat menambahkan instance berbasis waktu baru ke lapisan dengan membuka halaman berbasis Waktu, dan memilih Tambahkan instance berbasis waktu (jika Anda belum menambahkan instance berbasis waktu ke lapisan) atau + Instance (jika lapisan sudah memiliki satu atau lebih instance berbasis waktu). Kemudian, konfigurasikan instance seperti yang dijelaskan dalam prosedur sebelumnya.

Menggunakan penskalaan berbasis beban otomatis

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Instans berbasis beban memungkinkan Anda memulai atau menghentikan instance dengan cepat sebagai respons terhadap perubahan lalu lintas masuk. AWS OpsWorks Stacks menggunakan CloudWatch data Amazon untuk menghitung metrik berikut untuk setiap lapisan, yang mewakili nilai rata-rata di semua instance lapisan:

- CPU: Konsumsi CPU rata-rata, seperti 80%
- Memori: Konsumsi memori rata-rata, seperti 60%
- Beban: Rata-rata pekerjaan komputasi yang dilakukan sistem dalam satu menit.

Anda menentukan ambang batas peningkatan dan penurunan skala untuk salah satu atau semua metrik ini. Anda juga dapat menggunakan CloudWatch alarm khusus sebagai ambang batas.

Melintasi ambang batas memicu peristiwa penskalaan. Anda menentukan bagaimana AWS OpsWorks Stacks merespons peristiwa penskalaan dengan menentukan hal berikut:

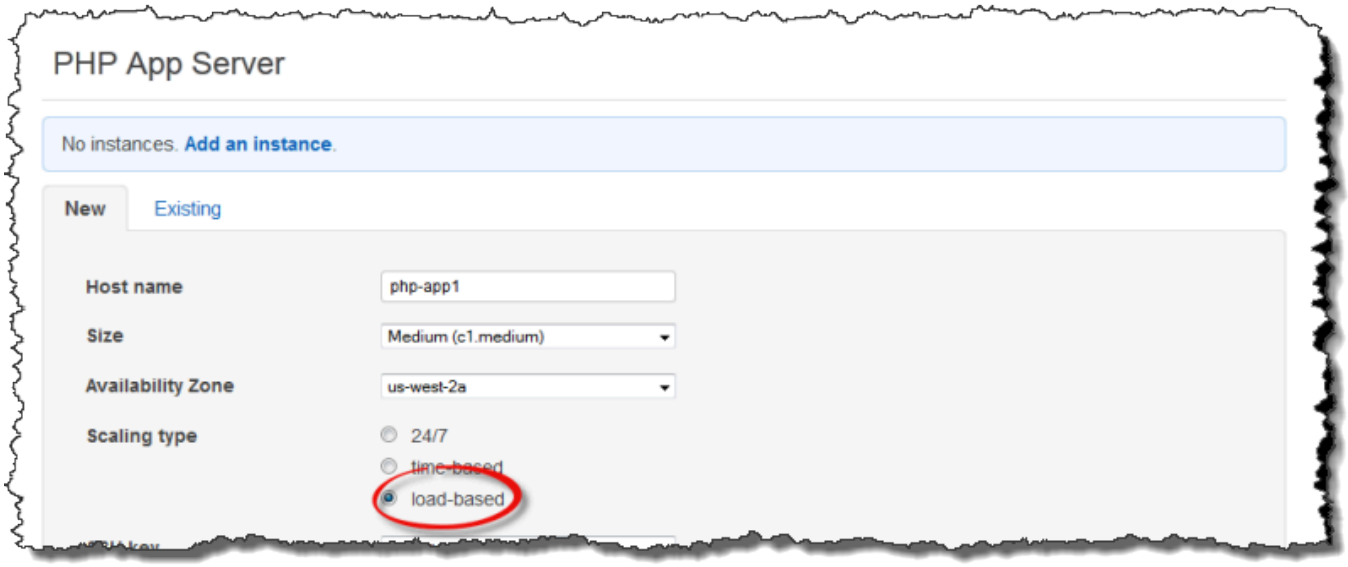
- Berapa banyak contoh untuk memulai atau berhenti.
- Berapa lama AWS OpsWorks Stacks harus menunggu setelah melebihi ambang batas sebelum memulai atau menghapus instance. Misalnya, pemanfaatan CPU harus melebihi ambang batas setidaknya selama 15 menit. Nilai ini memungkinkan Anda untuk mengabaikan fluktuasi lalu lintas singkat.
- Berapa lama AWS OpsWorks Stacks harus menunggu setelah memulai atau menghentikan instance sebelum memantau metrik lagi. Anda biasanya ingin memberikan waktu yang cukup untuk instance yang dimulai untuk online atau menghentikan instance untuk dimatikan sebelum menilai apakah lapisan tersebut masih melebihi ambang batas.

Ketika peristiwa penskalaan terjadi, AWS OpsWorks Stacks memulai atau menghentikan hanya instance berbasis beban. Itu tidak memulai atau menghentikan instance 24/7 atau instance berbasis waktu.

> **Note**
>
> Penskalaan berbasis beban otomatis tidak membuat instance baru; itu memulai dan menghentikan hanya instance yang telah Anda buat. Oleh karena itu, Anda harus menyediakan instans berbasis beban yang cukup terlebih dahulu untuk menangani beban maksimum yang diantisipasi.

Untuk membuat instance berbasis beban

1. Pada halaman Instances, pilih +Instance untuk menambahkan instance. Pilih Advanced, lalu pilih load-based.



2. Konfigurasikan instance, lalu pilih Add Instance untuk menambahkan instance ke layer.

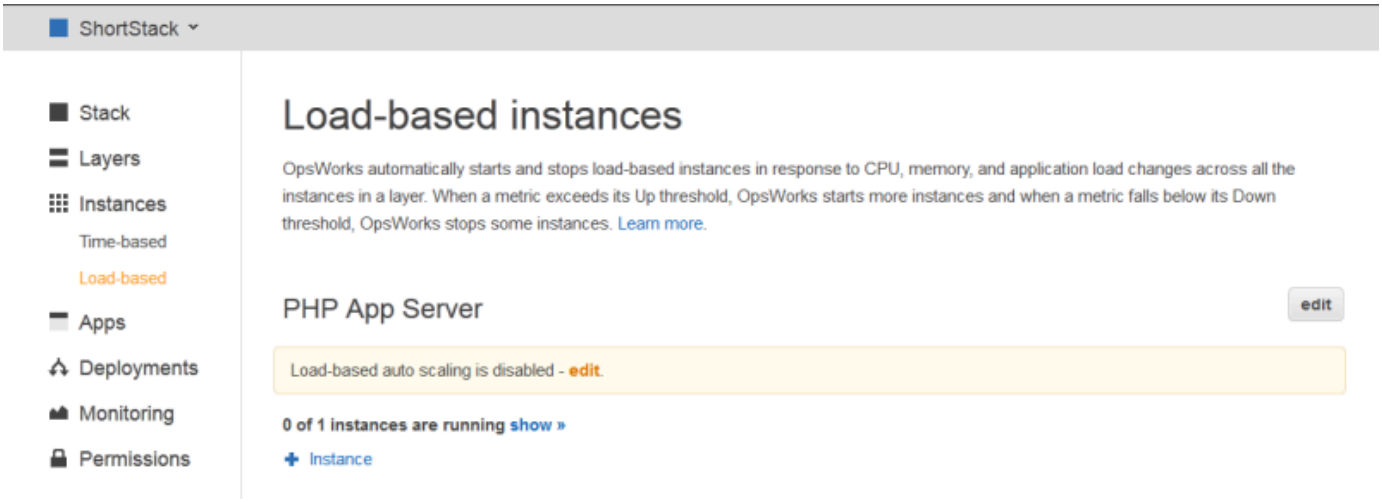
Ulangi prosedur ini sampai Anda telah membuat jumlah instance yang cukup. Anda dapat menambah atau menghapus instance nanti, seperti yang diperlukan.

Setelah menambahkan instance berbasis beban ke lapisan, Anda harus mengaktifkan penskalaan berbasis beban dan menentukan konfigurasi. Konfigurasi penskalaan berbasis beban adalah properti lapisan, bukan properti instance, yang menentukan kapan lapisan harus memulai atau

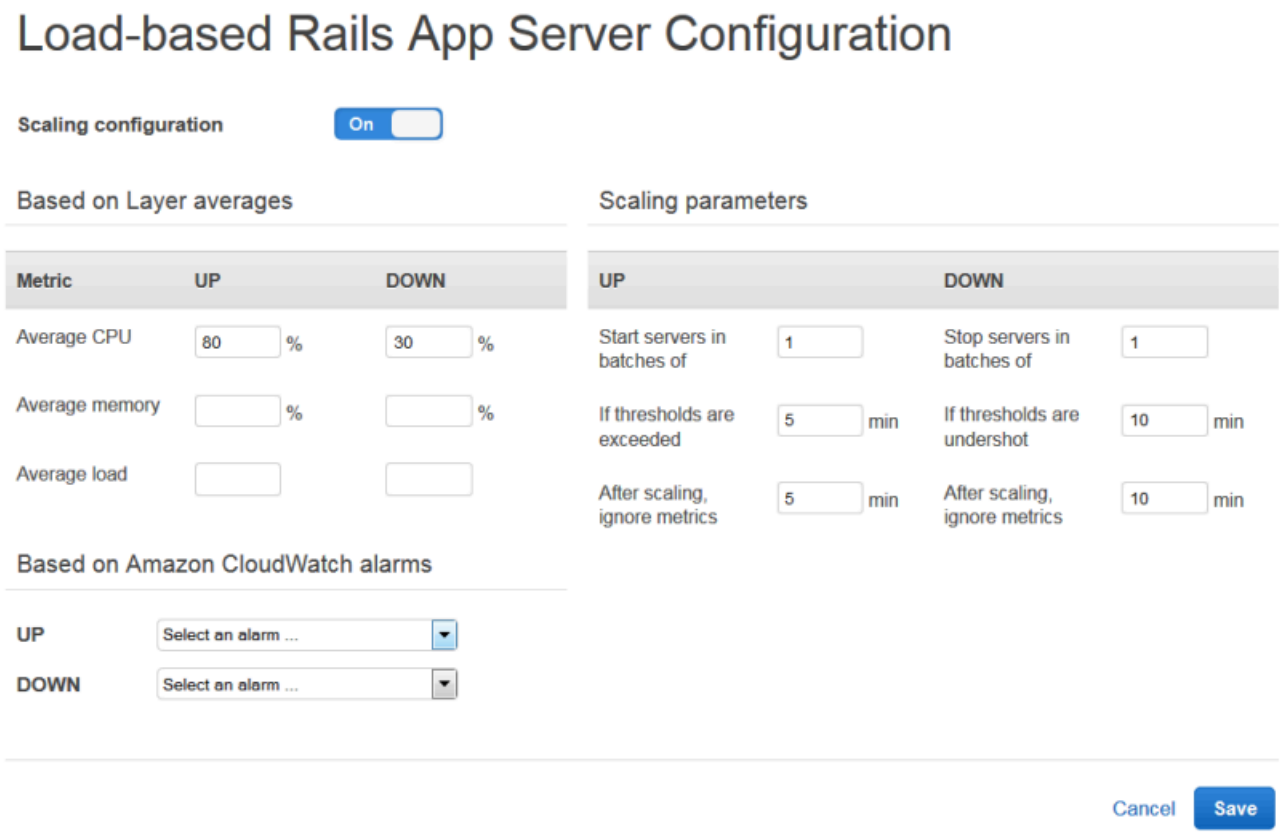
menghentikan instance berbasis beban. Ini harus ditentukan secara terpisah untuk setiap lapisan yang menggunakan instance berbasis beban.

Untuk mengaktifkan dan mengonfigurasi penskalaan berbasis beban otomatis

1. Di panel navigasi, di bawah Instans, pilih Berbasis beban, lalu pilih edit untuk lapisan yang sesuai.



2. Setel penskalaan otomatis berbasis beban yang diaktifkan ke Aktif. Kemudian atur parameter ambang batas dan penskalaan untuk menentukan bagaimana dan kapan harus menambah atau menghapus instance.



Ambang batas rata-rata lapisan

Anda dapat mengatur ambang penskalaan berdasarkan nilai berikut, yang dirata-ratakan pada semua instance layer.

- CPU rata-rata — Pemanfaatan CPU rata-rata lapisan, sebagai persen dari total.
- Memori rata-rata — Pemanfaatan memori rata-rata lapisan, sebagai persen dari total.
- Beban rata-rata - Beban rata-rata lapisan.

  Untuk informasi selengkapnya tentang bagaimana beban dihitung, lihat Memuat (komputasi) di Wikipedia.

Melintasi ambang batas menyebabkan peristiwa penskalaan, peningkatan jika lebih banyak instance diperlukan, dan downscaling jika lebih sedikit instance yang diperlukan. AWS OpsWorks Tumpukan kemudian menambahkan atau menghapus instance berdasarkan parameter penskalaan.

CloudWatch Alarm khusus

Anda dapat menggunakan hingga lima CloudWatch alarm khusus sebagai ambang batas peningkatan atau penurunan skala. Mereka harus berada di wilayah yang sama dengan

tumpukan. Untuk informasi selengkapnya tentang cara membuat alarm khusus, lihat Membuat CloudWatch Alarm Amazon.

> **Note**
>
> Untuk menggunakan alarm khusus, Anda harus memperbarui peran layanan Anda untuk mengizinkan`cloudwatch:DescribeAlarms`. Anda dapat meminta AWS OpsWorks Stacks memperbarui peran untuk Anda saat pertama kali menggunakan fitur ini, atau Anda dapat mengedit peran secara manual. Untuk informasi selengkapnya, lihat Mengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda.
>
> Ketika ada beberapa alarm yang dikonfigurasi untuk konfigurasi berbasis beban, jika alarm berada dalam status alarm `INSUFFICIENT_DATA` metrik, penskalaan instance berbasis beban tidak dapat terjadi meskipun alarm lain dalam status. `ALARM` Penskalaan otomatis hanya dapat dilanjutkan jika semua alarm berada di `OK` atau `ALARM` status. Untuk informasi selengkapnya tentang menggunakan CloudWatch alarm Amazon, lihat Menggunakan CloudWatch alarm Amazon di CloudWatch Panduan Pengguna Amazon.

Parameter penskalaan

Parameter berikut mengontrol cara AWS OpsWorks Stacks mengelola peristiwa penskalaan.

- Mulai server dalam batch — Jumlah instance yang akan ditambahkan atau dihapus saat peristiwa penskalaan terjadi.
- Jika ambang batas terlampaui — Jumlah waktu (dalam menit), beban harus tetap berada di atas ambang peningkatan atau di bawah ambang downscaling sebelum Stacks memicu peristiwa penskalaan. AWS OpsWorks
- Setelah penskalaan, abaikan metrik — Jumlah waktu (dalam menit) setelah peristiwa penskalaan terjadi bahwa AWS OpsWorks Stacks harus mengabaikan metrik dan menekan peristiwa penskalaan tambahan.

  Misalnya, AWS OpsWorks Stacks menambahkan instance baru setelah peristiwa upscaling tetapi instance tidak akan mulai mengurangi beban hingga di-boot dan dikonfigurasi. Tidak ada gunanya meningkatkan peristiwa penskalaan tambahan sampai instance baru online dan menangani permintaan, yang biasanya memakan waktu beberapa menit. Pengaturan

ini memungkinkan Anda mengarahkan AWS OpsWorks Stacks untuk menekan peristiwa penskalaan cukup lama untuk membuat instance baru online.

Anda dapat meningkatkan pengaturan ini untuk mencegah perubahan mendadak dalam penskalaan saat rata-rata lapisan seperti CPU Rata-rata, Memori rata-rata, atau beban rata-rata berada dalam ketidaksepakatan sementara.

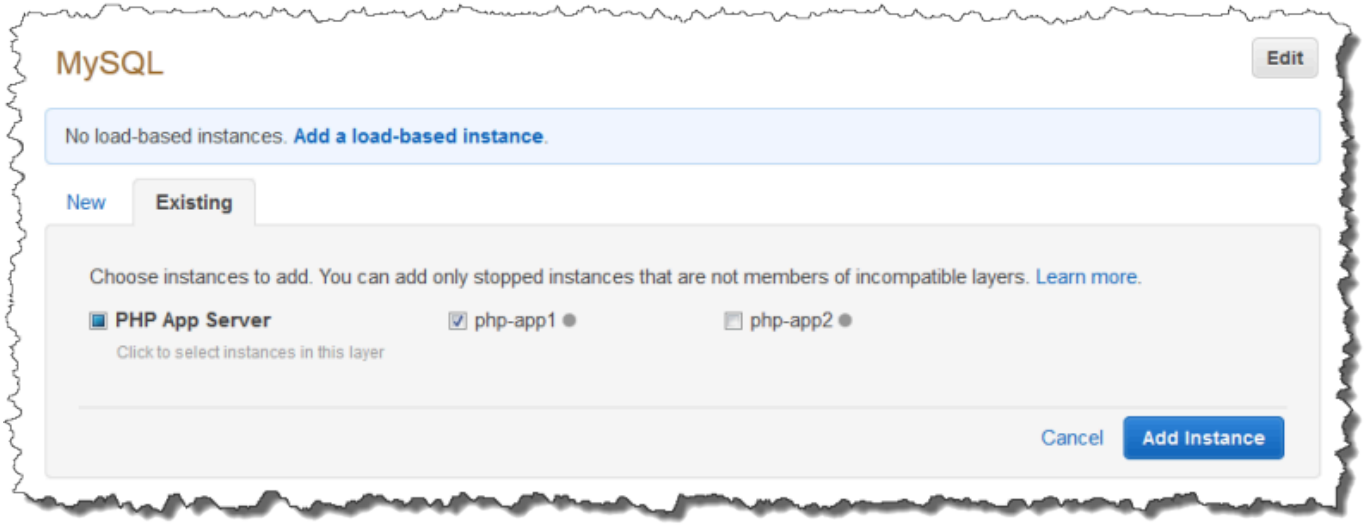
Misalnya, jika penggunaan CPU di atas batas dan penggunaan memori mendekati downscaling, peristiwa kelas atas instance mungkin segera diikuti oleh peristiwa downscaling memori. Untuk mencegah hal ini, Anda dapat menambah jumlah menit dalam pengaturan Setelah penskalaan, abaikan metrik. Dalam contoh ini, penskalaan CPU akan terjadi, tetapi peristiwa downscaling memori tidak akan terjadi.

3. Untuk menambahkan instance berbasis beban tambahan, pilih + Instance, konfigurasikan pengaturan, lalu pilih Tambah Instance. Ulangi sampai Anda memiliki cukup instans berbasis beban untuk menangani beban maksimum yang diantisipasi. Lalu, pilih Simpan.

> **Note**
>
> Anda juga dapat menambahkan instance berbasis beban baru ke lapisan dengan membuka halaman berbasis Beban, dan memilih Tambahkan instance berbasis beban (jika Anda belum menambahkan instance berbasis beban ke lapisan) atau + Instance (jika layer sudah memiliki satu atau lebih instance berbasis beban). Kemudian konfigurasikan instance seperti yang dijelaskan sebelumnya di bagian ini.

Untuk menambahkan instance berbasis beban yang ada ke lapisan

1. Di panel navigasi, di bawah Instans, pilih Berbasis beban.
2. Jika Anda telah mengaktifkan penskalaan otomatis berbasis beban untuk sebuah lapisan, pilih + Instance. Jika tidak, pilih Tambahkan instance berbasis beban. Pilih tab yang ada.



3. Pada tab yang ada, pilih sebuah instance. Daftar ini hanya menampilkan instance berbasis beban.

> **Note**
>
> Jika Anda berubah pikiran tentang menggunakan instance yang ada, pada tab Baru, buat instance baru seperti yang dijelaskan dalam prosedur sebelumnya.

4. Pilih Add Instance untuk menambahkan instance ke layer.

Anda dapat memodifikasi konfigurasi untuk atau menonaktifkan penskalaan berbasis beban otomatis kapan saja.

Untuk menonaktifkan penskalaan berbasis beban otomatis

1. Di panel navigasi, di bawah Instans, pilih Berbasis beban, lalu pilih edit untuk lapisan yang sesuai.
2. Alihkan penskalaan otomatis berbasis Beban yang diaktifkan ke No.

Bagaimana penskalaan berbasis beban berbeda dari penyembuhan otomatis

Penskalaan berbasis beban otomatis menggunakan metrik beban yang dirata-ratakan di semua instance yang sedang berjalan. Jika metrik tetap berada di antara ambang batas yang ditentukan, AWS OpsWorks Tumpukan tidak memulai atau menghentikan instance apa pun. Dengan auto healing, di sisi lain, AWS OpsWorks Stacks secara otomatis memulai instance baru dengan

konfigurasi yang sama ketika sebuah instance berhenti merespons. Instance mungkin tidak dapat merespons karena masalah jaringan atau masalah dengan instance.

Misalnya, anggaplah ambang peningkatan CPU Anda adalah 80%, dan satu instance berhenti merespons.

- Jika auto healing dinonaktifkan, dan instance yang berjalan lainnya dapat menjaga penggunaan CPU rata-rata di bawah 80%, AWS OpsWorks Stacks tidak memulai instance baru. Ini memulai instance pengganti hanya jika penggunaan CPU rata-rata di seluruh instance yang tersisa melebihi 80%.
- Jika penyembuhan otomatis diaktifkan, AWS OpsWorks Stacks memulai instance pengganti terlepas dari ambang beban.

Menggunakan Sumber Daya Komputasi yang Dibuat Di Luar AWS OpsWorks Tumpukan

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Fitur ini hanya didukung untuk tumpukan Linux.

Instansmenjelaskan cara menggunakan AWS OpsWorks Stacks untuk membuat dan mengelola grup instans Amazon Elastic Compute Cloud EC2 (Amazon). Anda juga dapat menggabungkan sumber daya komputasi Linux ke dalam tumpukan yang dibuat di luar AWS OpsWorks Stacks:

- EC2 Instans Amazon yang Anda buat langsung dengan menggunakan EC2 konsol Amazon, CLI, atau API.

- Instans lokal yang berjalan di perangkat keras Anda sendiri, termasuk instance yang berjalan di mesin virtual.

Sumber daya komputasi ini menjadi instans yang AWS OpsWorks dikelola Stacks, dan Anda dapat mengelolanya seperti instance Stacks biasa: AWS OpsWorks

- Kelola izin pengguna — Anda dapat menggunakan manajemen pengguna AWS OpsWorks Stacks untuk menentukan pengguna mana yang diizinkan mengakses tumpukan Anda, tindakan mana yang diizinkan untuk dilakukan pada instance tumpukan, dan apakah mereka memiliki akses SSH dan hak istimewa sudo.
- Mengotomatiskan tugas — Anda dapat AWS OpsWorks meminta Stacks menjalankan resep Chef khusus untuk melakukan tugas-tugas seperti menjalankan skrip pada salah satu atau semua instance tumpukan dengan satu perintah.

  Jika Anda menetapkan instance ke lapisan, AWS OpsWorks Stacks secara otomatis menjalankan serangkaian resep Chef tertentu pada instance pada titik-titik penting dalam siklus hidupnya, termasuk resep kustom Anda. Perhatikan bahwa Anda dapat menetapkan EC2 instans Amazon terdaftar ke lapisan khusus saja.
- Mengelola sumber daya — Tumpukan memungkinkan Anda mengelompokkan dan mengelola sumber daya dalam sebuah Wilayah AWS, dan OpsWorks dasbor menunjukkan status tumpukan Anda di semua Wilayah.
- Instal paket — Anda dapat menggunakan resep Chef untuk menginstal paket pada instance apa pun dalam tumpukan.
- Perbarui sistem operasi — AWS OpsWorks Stacks menyediakan cara sederhana untuk menginstal patch keamanan sistem operasi dan pembaruan pada instance tumpukan.
- Menyebarkan aplikasi — AWS OpsWorks Stacks menyebarkan aplikasi secara konsisten ke semua instance server aplikasi stack.
- Monitoring - AWS OpsWorks Stacks membuat CloudWatchmetrik khusus untuk memantau semua instance tumpukan Anda.

Untuk informasi harga, lihat OpsWorks Harga AWS.

Berikut ini adalah prosedur dasar untuk bekerja dengan instance terdaftar.

1. Daftarkan instance dengan tumpukan.

Instance sekarang menjadi bagian dari tumpukan dan dikelola oleh AWS OpsWorks Stacks.

2. Secara opsional, tetapkan instance ke lapisan.

   Langkah ini memungkinkan Anda memanfaatkan sepenuhnya fungsionalitas manajemen AWS OpsWorks Stacks. Anda dapat menetapkan instans lokal terdaftar ke lapisan apa pun; instans Amazon terdaftar EC2 hanya dapat ditetapkan ke lapisan khusus.
3. Gunakan AWS OpsWorks Stacks untuk mengelola instance.
4. Saat Anda tidak lagi membutuhkan instance di tumpukan, batalkan pendaftarannya, yang menghapus instance dari Stacks. AWS OpsWorks

Bagian berikut menjelaskan proses ini secara rinci.

Topik

- Mendaftarkan Instance dengan AWS OpsWorks Stacks Stack
- Mengelola Instans Terdaftar
- Menetapkan Instance Terdaftar ke Layer
- Membatalkan Penetapan Instance Terdaftar
- Menderegistrasi Instance Terdaftar
- Siklus Hidup Instans Terdaftar

Mendaftarkan Instance dengan AWS OpsWorks Stacks Stack

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Fitur ini hanya didukung untuk tumpukan Linux.

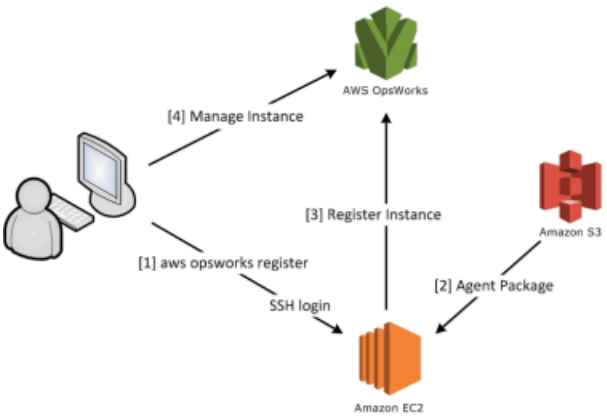
Untuk mendaftarkan instance yang berada di luar AWS OpsWorks Stacks, Anda menjalankan AWS CLI aws opsworks register perintah. Anda dapat menjalankan perintah ini dari instance yang ingin Anda daftarkan, atau dari komputer lain. Anda menerapkan `AWSOpsWorksRegisterCLI_OnPremises` kebijakan `AWSOpsWorksRegisterCLI_EC2` atau ke pengguna atau grup untuk memberikan izin yang diperlukan AWS CLI untuk mendaftar EC2 atau instans lokal, masing-masing. Kebijakan ini memerlukan versi 1.16.180 AWS CLI atau yang lebih baru.

> **Note**
>
> Untuk mencegah pengguna atau peran mendaftarkan instance, perbarui profil instans untuk menolak akses ke register perintah.

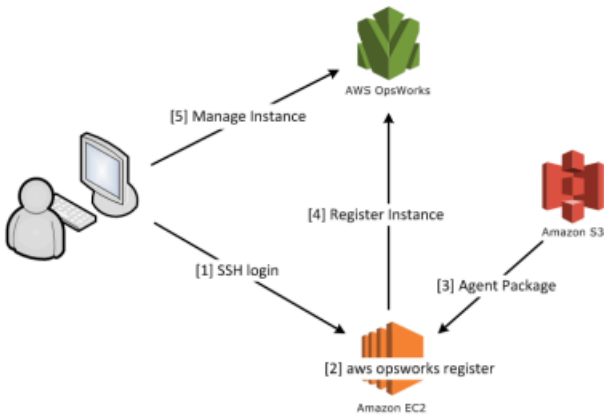
Proses pendaftaran menginstal agen pada instance yang ingin Anda kelola dengan menggunakan AWS OpsWorks Stacks, dan mendaftarkan instance dengan AWS OpsWorks tumpukan yang Anda tentukan. Setelah Anda mendaftarkan instance, instance adalah bagian dari tumpukan dan dikelola oleh AWS OpsWorks Stacks. Untuk informasi selengkapnya, lihat Mengelola Instans Terdaftar.

> **Note**
>
> Meskipun AWS Tools untuk PowerShell menyertakan `Register-OpsInstance`cmdlet, yang memanggil tindakan `register` API, kami menyarankan Anda menggunakan AWS CLI untuk menjalankan `register` perintah sebagai gantinya.

Diagram berikut menunjukkan kedua pendekatan untuk mendaftarkan EC2 instance Amazon. Anda dapat menggunakan pendekatan yang sama untuk mendaftarkan instans lokal.



Registering an instance from a workstation



Registering an instance from the instance itself

> **Note**
>
> Anda dapat menggunakan konsol AWS OpsWorks Stacks untuk mengelola instans terdaftar, tetapi Anda harus menjalankan perintah AWS `register` CLI untuk mendaftarkan instance. Alasan untuk ini adalah bahwa proses pendaftaran harus dijalankan dari instance, yang tidak dapat dilakukan oleh konsol.

Bagian berikut menjelaskan prosedur secara rinci.

Topik

- Walkthrough: Daftarkan Instance dari Workstation Anda
- Mendaftarkan Amazon EC2 dan Instans Lokal

Walkthrough: Daftarkan Instance dari Workstation Anda

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Proses pendaftaran mendukung beberapa skenario. Bagian ini memandu Anda melalui end-to-end contoh satu skenario: cara menggunakan workstation Anda untuk mendaftarkan EC2 instans Amazon. Skenario pendaftaran lainnya menggunakan prosedur serupa. Untuk informasi selengkapnya, lihat Mendaftarkan Amazon EC2 dan Instans Lokal.

> **Note**
>
> Anda biasanya ingin mendaftarkan EC2 instans Amazon yang ada. Namun, Anda bisa membuat instance baru dan tumpukan baru untuk panduan dan menghapusnya setelah selesai.

Topik

- Langkah 1: Buat Stack dan Instance
- Langkah 2: Instal dan Konfigurasikan AWS CLI
- Langkah 3: Daftarkan Instance dengan EC2 Register Stack

Langkah 1: Buat Stack dan Instance

Untuk memulai, Anda memerlukan tumpukan dan EC2 instance Amazon untuk didaftarkan dengan tumpukan itu.

Untuk membuat tumpukan dan instance

1. Gunakan konsol AWS OpsWorks Stacks untuk membuat tumpukan baru bernama**`EC2Register`**. Anda dapat menerima nilai default untuk pengaturan tumpukan lainnya.
2. Luncurkan instance baru dari EC2 konsol Amazon. Perhatikan hal berikut.

   - Instance harus berada di wilayah dan VPC yang sama dengan tumpukan.

     Jika Anda menggunakan VPC, pilih subnet publik untuk panduan ini.
   - Jika Anda perlu membuat kunci SSH, simpan file kunci pribadi ke workstation Anda dan catat nama dan lokasi file.

Jika Anda menggunakan kunci yang ada, catat nama dan lokasi file kunci pribadi. Anda membutuhkan nilai-nilai itu nanti.

- Instans harus didasarkan pada salah satu sistem operasi Linux yang didukung. Misalnya, jika tumpukan Anda berada di US West (Oregon), Anda dapat menggunakan `ami-35501205` untuk meluncurkan instance Ubuntu 14.04 LTS di wilayah itu.

Jika tidak, terima nilai default.

Saat instance booting, Anda dapat melanjutkan ke bagian berikutnya.

Langkah 2: Instal dan Konfigurasikan AWS CLI

Pendaftaran dilakukan dengan menggunakan AWS CLI aws opsworks register perintah. Sebelum Anda mendaftarkan instans pertama Anda, Anda harus menjalankan versi 1.16.180 AWS CLI atau yang lebih baru. Detail instalasi tergantung pada sistem operasi workstation Anda. Untuk informasi selengkapnya tentang menginstal AWS CLI, lihat Menginstal AWS Command Line Interface. Untuk memeriksa versi AWS CLI yang Anda jalankan, masukkan `aws --version` dalam sesi shell.

> **Note**
>
> Untuk mencegah pengguna atau peran mendaftarkan instance, perbarui profil instans untuk menolak akses ke register perintah.

Kami sangat menyarankan agar Anda tidak melewatkan langkah ini, bahkan jika Anda sudah menjalankan AWS CLI di workstation Anda. Menggunakan rilis terbaru dari ini AWS CLI adalah praktik terbaik keamanan.

Anda harus `register` menyediakan satu set kredensi AWS yang memiliki izin yang sesuai. Cara yang disarankan untuk melakukan ini—sehingga Anda menghindari menginstal kredensional secara langsung pada sebuah instance—adalah dengan mendaftarkan instance yang diluncurkan dengan profil instans, dan kemudian menambahkan sakelar ke perintah Anda. `--use-instance-profile register` Jika Anda mendapatkan kredensi dari profil instance, lewati ke topik Langkah 3: Daftarkan Instance dengan EC2 Register Stack ini. Namun, jika instans Anda tidak diluncurkan dengan profil instans, Anda dapat membuat pengguna IAM. Prosedur berikut membuat pengguna baru dengan izin yang sesuai, menginstal kredensyal pengguna di workstation, dan kemudian meneruskan kredensyal tersebut ke. `register`

> **Warning**
>
> Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Untuk membuat pengguna

1. Di konsol IAM, pilih Pengguna di panel navigasi, lalu pilih Tambah pengguna.
2. Tambahkan pengguna bernama**`EC2Register`**.
3. Pilih Berikutnya.
4. Pada halaman Setel izin, pilih Lampirkan kebijakan secara langsung.
5. Masukkan **`OpsWorks`** kotak Filter kebijakan izin untuk menampilkan AWS OpsWorks kebijakan, pilih salah satu kebijakan berikut, lalu pilih Berikutnya: tinjau. Kebijakan ini memberi pengguna izin yang diperlukan untuk menjalankannya. `register`
   - Pilih `AWSOpsWorksRegisterCLI_EC2` untuk mengizinkan izin pengguna untuk mendaftarkan EC2 instance yang menggunakan profil instans.
   - Pilih `AWSOpsWorksRegisterCLI_OnPremises` untuk mengizinkan izin pengguna untuk mendaftarkan instance lokal.
6. Pilih Berikutnya.
7. Pada halaman Tinjauan, pilih Buat pengguna.
8. Sekarang buat kunci akses untuk pengguna Anda. Dari panel navigasi, pilih Pengguna, lalu pilih pengguna yang ingin Anda buat kunci aksesnya.
9. Pilih tab Security credentials, lalu pilih Create access key.
10. Pilih praktik & alternatif terbaik kunci Akses yang paling sesuai dengan tugas Anda.
11. Pilih Berikutnya.
12. (Opsional) masukkan tag untuk mengidentifikasi kunci akses.
13. Pilih Berikutnya.
14. Pilih Unduh file.csv, simpan file kredensialnya ke lokasi yang nyaman di sistem Anda, dan pilih Selesai.

Anda perlu memberikan kredensi pengguna IAM ke. `register` Panduan ini menangani tugas dengan menginstal kredensyal EC2 Register di file workstation Anda. `credentials` Untuk informasi tentang cara lain mengelola kredensialnya AWS CLI, lihat File Konfigurasi dan Kredensyal.

Untuk menginstal kredensi pengguna

1. Buat atau buka `credentials` file workstation Anda. File ini terletak di `~/.aws/credentials` (Linux, Unix, dan OS X) atau `C:\Users\`*`User_Name`*`\.aws\credentials` (sistem Windows).
2. Tambahkan profil untuk pengguna EC2 Register ke `credentials` file, menggunakan format berikut.

   ```
   [ec2register]
   aws_access_key_id = access_key_id
   aws_secret_access_key = secret_access_key
   ```

   Ganti *`access_key_id`* dan *`secret_access_key`* dengan tombol EC2 Daftar untuk yang Anda unduh sebelumnya.

Langkah 3: Daftarkan Instance dengan EC2 Register Stack

Anda sekarang siap untuk mendaftarkan instance.

Untuk mendaftarkan instance

1. Di AWS OpsWorks Tumpukan, kembali ke tumpukan EC2 Register, pilih Instans di panel navigasi, lalu pilih Daftar instance.
2. Pilih EC2 Instans, pilih Berikutnya: Pilih Instans, dan pilih instans Anda dari daftar.
3. Pilih Berikutnya: Instal AWS CLI, dan Berikutnya: Daftarkan Instans. AWS OpsWorks Stacks secara otomatis menggunakan informasi yang tersedia, seperti ID tumpukan dan ID instance untuk membuat template `register` perintah, yang ditampilkan di halaman Daftar Instans. Untuk contoh ini, Anda gunakan **`register`** untuk masuk ke instance dengan kunci SSH dan secara eksplisit menentukan file kunci, jadi atur saya menggunakan kunci SSH untuk terhubung ke instance saya ke Ya. Template perintah menyerupai berikut ini.

   ```
   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**
>
> Anda harus menyetel Wilayah ke Wilayah titik akhir layanan AWS OpsWorks Stacks, bukan Wilayah tumpukan, jika tumpukan berada dalam Wilayah klasik yang terkait dengan titik akhir `us-east-1` regional. AWS OpsWorks Tumpukan menentukan Wilayah tumpukan dari ID tumpukan.

4. Template perintah berisi beberapa nilai argumen khusus pengguna, yang ditunjukkan oleh tanda kurung dan harus diganti dengan nilai yang sesuai. Salin template perintah ke editor teks dan edit sebagai berikut.

> **Important**
>
> Pengguna IAM yang dibuat selama proses pendaftaran diperlukan sepanjang masa pakai instance terdaftar. Menghapus pengguna menyebabkan agen AWS OpsWorks Stacks tidak dapat berkomunikasi dengan layanan. Untuk membantu mencegah masalah dalam mengelola instance terdaftar jika pengguna terhapus secara tidak sengaja, tambahkan `--use-instance-profile` parameter ke `register` perintah Anda untuk menggunakan profil instans bawaan instans sebagai gantinya. Menambahkan `--use-instance-profile` parameter juga mencegah kesalahan terjadi ketika Anda memutar kunci akses AWS akun setiap 90 hari (praktik terbaik yang disarankan), karena mencegah ketidakcocokan antara kunci akses yang tersedia untuk AWS OpsWorks agen dan pengguna IAM yang diperlukan.

- Ganti *`key file`* dengan path yang sepenuhnya memenuhi syarat dari file kunci pribadi untuk EC2 key pair Amazon yang disimpan saat membuat instance.

  Anda dapat menggunakan jalur relatif, jika Anda mau.

- Ganti *`username`* dengan nama pengguna instance.

  Untuk contoh ini, nama pengguna adalah`ubuntu`, untuk instance Ubuntu, atau`ec2-user`, untuk Red Hat Enterprise Linux (RHEL) atau instance Amazon Linux.

- Add`--use-instance-profile`, yang berjalan `register` dengan profil instance untuk mencegah kesalahan selama rotasi kunci atau jika pengguna IAM utama terhapus secara tidak sengaja.

Perintah Anda harus menyerupai yang berikut ini.

```
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. Buka jendela terminal di workstation Anda, tempel `register` perintah dari editor Anda, dan jalankan perintah.

   Pendaftaran biasanya memakan waktu sekitar lima menit. Setelah selesai, kembali ke konsol AWS OpsWorks Stacks dan pilih Selesai. Kemudian pilih Instans di panel navigasi. Instance Anda harus terdaftar di bawah Instans yang Tidak Ditugaskan. Anda kemudian dapat menetapkan instance ke lapisan atau membiarkannya di tempatnya, tergantung pada bagaimana Anda bermaksud mengelola instance.
6. Setelah selesai, hentikan instance dan hapus dengan menggunakan konsol AWS OpsWorks Stacks atau perintah. Ini mengakhiri EC2 instans Amazon, sehingga Anda tidak dikenakan biaya lebih lanjut.

Mendaftarkan Amazon EC2 dan Instans Lokal

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Bagian ini menjelaskan cara mendaftarkan Amazon EC2 atau instans lokal dengan AWS OpsWorks tumpukan Stacks.

Topik

- Mempersiapkan Instance
- Menginstal dan Mengkonfigurasi AWS CLI
- Mendaftarkan Instance
- Menggunakan register Command
- Contoh register Perintah
- Kebijakan Pendaftaran Instans

Mempersiapkan Instance

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Sebelum mendaftarkan instance, Anda harus memastikan bahwa itu kompatibel dengan AWS OpsWorks Stacks. Detailnya bergantung pada apakah Anda mendaftarkan EC2 instans lokal atau Amazon.

Instans Lokal

Instance lokal harus memenuhi kriteria berikut:

- Instance harus menjalankan salah satu sistem operasi Linux yang didukung. Meskipun dimungkinkan untuk membuat atau mendaftarkan instance dengan sistem operasi lain (seperti CentOS 6. x) yang telah dibuat dari kustom atau yang dihasilkan oleh komunitas AMIs, ini tidak didukung secara resmi.

Anda harus menginstal `libyaml` paket pada instance. Untuk contoh Ubuntu, paket diberi nama`libyaml-0-2`. Untuk CentOS dan Red Hat Enterprise Linux instance, paket ini diberi nama. `libyaml`

- Instance harus memiliki tipe instance yang didukung (kadang-kadang disebut ukuran instance). Jenis instans yang didukung dapat bervariasi menurut sistem operasi, dan bergantung pada apakah tumpukan Anda ada di VPC. Untuk daftar jenis instans yang didukung, lihat nilai daftar drop-down Ukuran yang ditampilkan di konsol AWS OpsWorks Stacks saat Anda mencoba membuat instance baru di tumpukan target. Jika jenis instance berwarna abu-abu, dan tidak dapat dibuat di tumpukan target Anda, maka Anda tidak dapat mendaftarkan instance dari jenis itu.
- Instance harus memiliki akses Internet yang memungkinkannya berkomunikasi dengan titik akhir layanan AWS OpsWorks Stacks,. `opsworks.us-east-1.amazonaws.com (HTTPS)` Instans juga harus mendukung koneksi keluar ke sumber daya AWS seperti Amazon S3.
- Jika Anda berencana untuk mendaftarkan instance dari workstation terpisah, instance terdaftar harus mendukung login SSH dari workstation.

  Login SSH tidak diperlukan jika Anda menjalankan perintah registrasi dari instance.
- Kunci AWS akses digunakan untuk otentikasi dari AWS OpsWorks agen ke layanan AWS OpsWorks Stacks. Jika Anda memutar tombol akses seperti yang disarankan setiap 90 hari, perbarui AWS OpsWorks agen secara manual untuk menggunakan kunci baru. Pada komputer atau instans lokal, edit `/etc/aws/opsworks/instance-agent.yml` file dengan kunci akses baru dan kunci rahasia. Perintah berikut menunjukkan kunci akses dan kunci rahasia dalam file ini. Agen yang menggunakan kunci lama dapat menyebabkan kesalahan.

```
cat /etc/aws/opsworks/instance-agent.yml | egrep "access_key|secret_key"
:access_key_id: AKIAIOSFODNN7EXAMPLE
:secret_access_key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

EC2Contoh Amazon

EC2 Instans Amazon harus memenuhi kriteria berikut:

- AMI harus didasarkan pada salah satu sistem operasi Linux yang didukung. Untuk daftar saat ini, lihatAWS OpsWorks Stacks sistem operasi.

  Untuk informasi selengkapnya, lihat Menggunakan Custom AMIs.

Jika instance didasarkan pada AMI kustom yang berasal dari AMI standar yang didukung, atau jika instance berisi pengaturan yang sangat minim, Anda harus menginstal `libyaml` paket pada instance. Untuk contoh Ubuntu, paket diberi nama`libyaml-0-2`. Untuk instans Amazon Linux dan Red Hat Enterprise Linux, paket ini diberi nama`libyaml`.

- Instance harus memiliki tipe instance yang didukung (kadang-kadang disebut ukuran instance). Jenis instans yang didukung dapat bervariasi menurut sistem operasi, dan bergantung pada apakah tumpukan Anda ada di VPC. Untuk daftar jenis instans yang didukung, lihat nilai daftar drop-down Ukuran yang ditampilkan di konsol AWS OpsWorks Stacks saat Anda mencoba membuat instance baru di tumpukan target. Jika tipe instance berwarna abu-abu, dan tidak dapat dibuat di tumpukan target Anda, maka Anda juga tidak dapat mendaftarkan instance dari jenis itu.
- Instans harus berada dalam status `running`.
- Instance tidak boleh menjadi bagian dari grup Auto Scaling.

  Untuk informasi selengkapnya, lihat Melepaskan EC2 Instans Dari Grup Auto Scaling Anda.
- Instance dapat menjadi bagian dari VPC, tetapi harus dalam VPC yang sama dengan tumpukan dan VPC harus dikonfigurasi agar berfungsi dengan baik dengan Stacks. AWS OpsWorks
- Instans spot tidak didukung, karena mereka tidak bekerja dengan auto healing.

Saat Anda mendaftarkan EC2 instans Amazon, AWS OpsWorks Stacks tidak mengubah grup atau aturan keamanan instans. Pastikan aturan grup keamanan instans sesuai dengan persyaratan AWS OpsWorks Stacks berikut.

Aturan Ingress

Aturan masuk harus memungkinkan hal berikut.

- Login SSH.
- Lalu lintas dari lapisan yang sesuai.

  Misalnya, server database biasanya akan memungkinkan lalu lintas masuk dari lapisan server aplikasi stack.
- Lalu lintas ke port yang sesuai.

  Misalnya, instance server aplikasi biasanya memungkinkan semua lalu lintas masuk ke port 80 (HTTP) dan 443 (HTTPS).

Aturan Jalan keluar

Aturan jalan keluar harus memungkinkan yang berikut ini.

- Lalu lintas ke layanan AWS OpsWorks Stacks dari aplikasi yang berjalan pada instance.
- Lalu lintas untuk mengakses sumber daya AWS seperti Amazon S3 dari aplikasi yang menggunakan AWS API.

Salah satu pendekatan umum adalah tidak menentukan aturan jalan keluar, yang tidak membatasi lalu lintas keluar.

Menginstal dan Mengkonfigurasi AWS CLI

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sebelum Anda mendaftarkan instance pertama Anda, Anda harus menjalankan versi 1.16.180 dari AWS CLI atau yang lebih baru di komputer tempat Anda menjalankan. `register` Detail instalasi tergantung pada sistem operasi workstation Anda. Untuk informasi selengkapnya tentang menginstal AWS CLI, lihat Menginstal AWS Command Line Interface dan Mengonfigurasi AWS Command Line Interface. Untuk memeriksa versi AWS CLI yang Anda jalankan, masukkan `aws --version` dalam sesi shell.

> **Note**
>
> Meskipun AWS Tools untuk PowerShell menyertakan `Register-OpsInstance`cmdlet, yang memanggil tindakan `register` API, kami menyarankan Anda menggunakan AWS CLI untuk menjalankan `register` perintah sebagai gantinya.

Anda harus menjalankan `register` dengan izin yang sesuai. Anda bisa mendapatkan izin dengan menggunakan peran IAM, atau kurang optimal, dengan menginstal kredensyal pengguna dengan

izin yang sesuai pada workstation atau instance yang akan didaftarkan. Anda kemudian dapat menjalankan `register` dengan kredensional tersebut, seperti yang dijelaskan nanti. Tentukan izin dengan melampirkan kebijakan IAM ke pengguna atau peran. Untuk`register`, Anda menggunakan `AWSOpsWorksRegisterCLI_OnPremises` kebijakan `AWSOpsWorksRegisterCLI_EC2` atau kebijakan, yang masing-masing memberikan izin untuk mendaftarkan Amazon EC2 atau instans lokal.

> **Note**
>
> Jika Anda menjalankan EC2 instans `register` Amazon, idealnya Anda harus menggunakan peran IAM untuk memberikan kredensi. Untuk informasi selengkapnya tentang cara melampirkan peran IAM ke instance yang ada, lihat Melampirkan peran IAM ke instance atau Mengganti peran IAM di Panduan Pengguna Amazon EC2 .

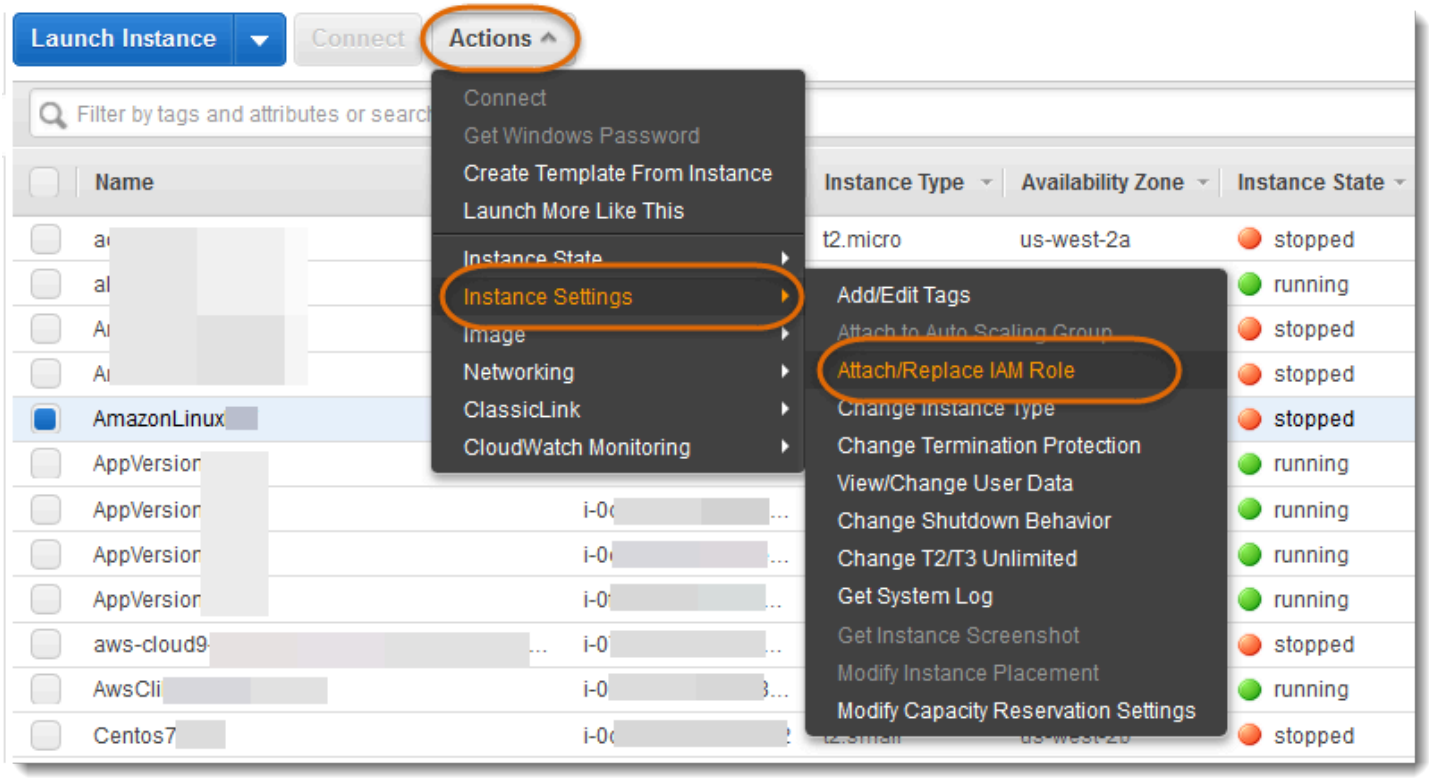
Misalnya cuplikan `AWSOpsWorksRegisterCLI_OnPremises` kebijakan `AWSOpsWorksRegisterCLI_EC2` dan kebijakan, lihat. Kebijakan Pendaftaran Instans Untuk informasi selengkapnya tentang membuat dan mengelola kredensi AWS, lihat AWS Security Credentials.

Topik

- Menggunakan Peran IAM
- Menggunakan Kredensial Terinstal

Menggunakan Peran IAM

Jika Anda menjalankan perintah dari EC2 instans Amazon yang ingin Anda daftarkan, strategi yang lebih disukai untuk memberikan kredensialnya `register` adalah dengan menggunakan peran IAM yang memiliki `AWSOpsWorksRegisterCLI_EC2` kebijakan atau izin setara yang dilampirkan. Pendekatan ini memungkinkan Anda untuk menghindari menginstal kredensional Anda pada instance. Salah satu cara untuk melakukannya adalah dengan menggunakan perintah Attach/ Replace IAM Role di EC2 konsol, seperti yang ditunjukkan pada gambar berikut.



Untuk informasi selengkapnya tentang cara melampirkan peran IAM ke instance yang ada, lihat Melampirkan peran IAM ke instance atau Mengganti peran IAM di Panduan Pengguna Amazon EC2 . Untuk contoh yang diluncurkan dengan profil instance (disarankan), tambahkan `--use-instance-profile` sakelar ke `register` perintah Anda untuk memberikan kredensyal; jangan gunakan parameter. `--profile`

Jika instance sedang berjalan dan memiliki peran, Anda dapat memberikan izin yang diperlukan dengan melampirkan `AWSOpsWorksRegisterCLI_EC2` kebijakan ke peran tersebut. Peran menyediakan satu set kredensi default untuk instance. Selama Anda belum menginstal kredensi apa pun pada instance, `register` secara otomatis mengasumsikan peran tersebut dan berjalan dengan izinnya.

> **Important**
>
> Kami menyarankan Anda untuk tidak menginstal kredensional pada instance. Selain menciptakan risiko keamanan, peran instans berada di akhir rantai penyedia default yang AWS CLI digunakan untuk menemukan kredensyal default. Kredensyal yang diinstal mungkin lebih diutamakan daripada peran, dan karena itu `register` mungkin tidak memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat Memulai dengan AWS CLI.

Jika instance yang sedang berjalan tidak memiliki peran, Anda harus menginstal kredensil dengan izin yang diperlukan pada instance, seperti yang dijelaskan dalam. Menggunakan Kredensial Terinstal Disarankan, lebih mudah, dan tidak terlalu rawan kesalahan untuk menggunakan instance yang diluncurkan dengan profil instance.

Menggunakan Kredensial Terinstal

Ada beberapa cara untuk menginstal kredensi pengguna pada suatu sistem dan memberikannya ke perintah AWS CLI . Berikut ini menjelaskan pendekatan yang tidak lagi direkomendasikan, tetapi dapat digunakan jika Anda mendaftarkan EC2 instance yang diluncurkan tanpa profil instance. Anda juga dapat menggunakan kredensi pengguna yang ada selama kebijakan terlampir memberikan izin yang diperlukan. Untuk informasi selengkapnya, termasuk deskripsi cara lain untuk menginstal kredensil, lihat File Konfigurasi dan Kredenal.

Untuk menggunakan kredensi yang diinstal

1. Buat pengguna IAM dan simpan ID kunci akses dan kunci akses rahasia di lokasi yang aman.

   > **Warning**
   >
   > Pengguna IAM memiliki kredensi jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan menghapus pengguna ini ketika mereka tidak lagi diperlukan.

2. Lampirkan AWSOpsWorksRegisterCLI_OnPremises kebijakan ke pengguna. Jika mau, Anda dapat melampirkan kebijakan yang memberikan izin yang lebih luas, asalkan menyertakan izin. `AWSOpsWorksRegisterCLI_OnPremises`

3. Buat profil untuk pengguna di `credentials` file sistem. File ini terletak di `~/.aws/credentials` (Linux, Unix, dan OS X) atau `C:\Users\`*`User_Name`*`\.aws\credentials` (sistem Windows). File berisi satu atau lebih profil dalam format berikut, yang masing-masing berisi ID kunci akses pengguna dan kunci akses rahasia.

   ```
   [profile_name]
   aws_access_key_id = access_key_id
   aws_secret_access_key = secret_access_key
   ```

Ganti kredenal IAM yang Anda simpan sebelumnya untuk nilai dan*access_key_id*. *secret_access_key* Anda dapat menentukan nama apa pun yang Anda inginkan untuk nama profil, dengan dua batasan: nama harus unik, dan profil default harus diberi nama`default`. Anda juga dapat menggunakan profil yang ada, asalkan memiliki izin yang diperlukan.

4. Gunakan `--profile` parameter `register` perintah untuk menentukan nama profil. `register`Perintah berjalan dengan izin yang diberikan ke kredensil terkait.

   Anda juga bisa menghilangkan`--profile`. Dalam hal ini, `register` berjalan dengan kredensil default. Ketahuilah bahwa ini belum tentu kredensil profil default, jadi Anda harus memastikan bahwa kredenal default memiliki izin yang diperlukan. Untuk informasi selengkapnya tentang cara AWS CLI menentukan kredensil default, lihat Mengonfigurasi Antarmuka Baris Perintah AWS.

Mendaftarkan Instance

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Anda mendaftarkan instance dengan menjalankan AWS CLI `register` perintah dari workstation Anda atau dari instance. Cara paling sederhana untuk menangani operasi adalah dengan menggunakan wizard pendaftaran konsol AWS OpsWorks Stacks, yang menyederhanakan proses pembuatan string perintah. Setelah Anda terbiasa dengan prosedur pendaftaran, Anda dapat melewati wizard jika Anda mau, dan menjalankan `register` perintah.

Berikut ini menjelaskan cara menggunakan wizard pendaftaran untuk mendaftarkan instance dengan tumpukan yang ada.

> **Note**
>
> Untuk mendaftarkan instance dengan tumpukan baru, Anda dapat melakukannya dengan memilih Daftar Instans di dasbor AWS OpsWorks Stacks. Ini memulai wizard yang identik dengan yang ada untuk tumpukan yang ada, kecuali untuk halaman tambahan yang mengonfigurasi tumpukan baru.

Untuk menggunakan wizard pendaftaran untuk mendaftarkan sebuah instance

1. Di konsol AWS OpsWorks Stacks, buat tumpukan atau buka tumpukan yang ada.
2. Pilih Instans di panel navigasi, lalu pilih daftarkan instance.
3. Pada halaman Pilih Jenis Instance, tentukan apakah Anda ingin mendaftarkan Amazon EC2 atau instans lokal:
   - Jika Anda mendaftarkan EC2 instans Amazon, pilih Berikutnya: Pilih Instans.
   - Jika Anda mendaftarkan instans lokal, pilih Berikutnya: Instal AWS CLI, lalu buka Langkah 5.
4. Jika Anda mendaftarkan EC2 instans Amazon, buka halaman Select Instances untuk memilih instans yang akan didaftarkan. AWS OpsWorks Stacks mengumpulkan informasi yang diperlukan untuk membangun perintah. Setelah selesai, pilih Next:Install AWS CLI.
5. Instance yang Anda rencanakan untuk dijalankan `register` harus menjalankan versi 1.16.180 AWS CLI atau yang lebih baru. Untuk menginstal atau memperbarui AWS CLI, halaman panduan pendaftaran menyediakan tautan ke instruksi instalasi dan konfigurasi. Setelah Anda memverifikasi AWS CLI instalasi, pilih apakah Anda menjalankan perintah dari instance yang akan didaftarkan atau dari workstation terpisah, lalu pilih Next: Register Instances.
6. Halaman Daftar Instans menampilkan template untuk string `register` perintah yang menggabungkan opsi yang Anda pilih. Misalnya, jika Anda mendaftarkan EC2 instans Amazon dari workstation terpisah, templat defaultnya menyerupai berikut ini.

```
aws opsworks register --infrastructure-class ec2 --region us-west-2
  --stack-id 247be7ea-3551-4177-9524-1ff804f453e3 --ssh-username [username] i-
f1245d10
```

> **Important**
>
> Pengguna IAM yang dibuat selama proses pendaftaran diperlukan sepanjang masa pakai instance terdaftar. Menghapus pengguna menyebabkan agen AWS OpsWorks Stacks tidak dapat berkomunikasi dengan layanan. Untuk membantu mencegah masalah dalam mengelola instance terdaftar jika pengguna terhapus secara tidak sengaja, tambahkan `--use-instance-profile` parameter ke `register` perintah Anda untuk menggunakan profil instans bawaan instans sebagai gantinya. Menambahkan `--use-instance-profile` parameter juga mencegah kesalahan terjadi ketika Anda memutar kunci akses AWS akun setiap 90 hari (praktik terbaik yang disarankan), karena mencegah ketidakcocokan antara kunci akses yang tersedia untuk AWS OpsWorks agen dan pengguna IAM yang diperlukan.

Jika Anda menyetel I use SSH keys ke Yes, AWS OpsWorks Stacks menambahkan `--ssh-private-key` argumen ke string, yang dapat Anda gunakan untuk menentukan file kunci SSH pribadi.

> **Note**
>
> Jika Anda `register` ingin masuk dengan kata sandi, atur saya menggunakan kunci SSH ke No. Saat Anda menjalankan`register`, Anda akan diminta untuk kata sandi.

Salin string ini ke editor teks. dan edit sesuai kebutuhan. Perhatikan hal berikut.

- Teks tanda kurung mewakili informasi yang harus Anda berikan, seperti lokasi file kunci SSH Anda.
- Template mengasumsikan bahwa Anda menjalankan `register` dengan kredensi AWS default. Jika tidak, tambahkan `--profile` argumen ke string perintah, dan tentukan nama profil kredensyal yang ingin Anda gunakan.

Untuk skenario lain, Anda mungkin perlu mengubah perintah lebih lanjut. Untuk penjelasan tentang `register` argumen yang tersedia dan cara alternatif untuk membangun string perintah, lihatMenggunakan `register` Command. Anda juga dapat menampilkan dokumentasi perintah

dengan menjalankan `aws opsworks help register` dari baris perintah. Untuk beberapa contoh string perintah, lihatContoh register Perintah.

7. Setelah Anda selesai mengedit string perintah, buka jendela terminal di workstation Anda atau gunakan SSH untuk masuk ke instance dan jalankan perintah. Seluruh operasi biasanya memakan waktu sekitar lima menit, di mana instance berada dalam keadaan Mendaftar.
8. Setelah operasi selesai, pilih Selesai. Instance sekarang dalam status Terdaftar dan terdaftar sebagai instance yang tidak ditetapkan pada halaman Instans tumpukan.

`register`Perintah melakukan hal berikut.

1. Jika `register` berjalan di workstation, perintah pertama menggunakan SSH untuk masuk ke instance yang akan didaftarkan.

   Sisa proses berlangsung pada instance, dan sama terlepas dari di mana Anda menjalankan perintah.
2. Unduh paket agen AWS OpsWorks Stacks dari Amazon S3.
3. Membongkar dan menginstal agen dan dependensinya, seperti AWS SDK for Ruby.
4. Menciptakan yang berikut:
   - Pengguna IAM yang mem-bootstrap agen dengan layanan AWS OpsWorks Stacks untuk menyediakan komunikasi yang aman.

     Izin pengguna hanya mengizinkan `opsworks:RegisterInstance` tindakan, dan mereka kedaluwarsa setelah 15 menit.
   - Grup IAM untuk tumpukan, yang berisi pengguna instans terdaftar.
5. Membuat key pair RSA dan mengirimkan kunci publik ke AWS OpsWorks Stacks.

   Key pair ini digunakan untuk mengenkripsi komunikasi antara agen dan AWS OpsWorks Stacks.
6. Mendaftarkan instance dengan AWS OpsWorks Stacks. Tumpukan kemudian menjalankan satu set resep penyiapan awal untuk mengonfigurasi instance, yang mencakup yang berikut ini.
   - Menimpa file host instance.

     Dengan mendaftarkan instance, Anda telah menyerahkan manajemen pengguna ke AWS OpsWorks Stacks, yang harus memiliki file host sendiri untuk mengontrol izin login SSH.
   - Untuk EC2 instans Amazon, penyiapan awal juga mencakup mendaftarkan volume Amazon EBS terlampir atau alamat IP Elastis dengan tumpukan.

Anda harus memastikan bahwa volume Amazon EBS tidak dipasang ke titik pemasangan yang dicadangkan, termasuk `/var/www` dan titik pemasangan apa pun yang dicadangkan oleh lapisan instans. Untuk informasi selengkapnya tentang mengelola sumber daya tumpukan, lihatManajemen Sumber Daya. Untuk informasi selengkapnya tentang titik pemasangan lapisan, lihatAWS OpsWorks Referensi Lapisan Tumpukan.

Untuk deskripsi lengkap tentang perubahan konfigurasi penyiapan awal, lihatPerubahan Konfigurasi Pengaturan Awal.

> **Note**
>
> Penyiapan awal tidak memperbarui sistem operasi instans terdaftar; Anda harus menangani tugas itu sendiri. Untuk informasi selengkapnya, lihat Mengelola Pembaruan Keamanan.

Menggunakan **`register`** Command

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Untuk mendaftarkan instance, pastikan Anda menjalankan setidaknya versi 1.16.180 dari. AWS CLI Berikut ini menunjukkan sintaks umum untuk `register` perintah.

```
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]
```

Argumen berikut adalah umum untuk semua AWS CLI perintah.

`--profile`

(Opsional) Nama profil kredensialnya. Jika Anda menghilangkan argumen ini, perintah berjalan dengan kredenal default Anda. Untuk informasi selengkapnya tentang cara AWS CLI menentukan kredensil default, lihat Mengonfigurasi Antarmuka Baris Perintah AWS.

`--region`

(Opsional) AWS OpsWorks Wilayah titik akhir layanan Stacks. Jangan disetel `--region` ke Wilayah tumpukan. AWS OpsWorks Tumpukan secara otomatis menentukan Wilayah tumpukan dari ID tumpukan.

> **Note**
>
> Jika Region default Anda sudah disetel, Anda dapat menghilangkan argumen ini. Untuk informasi selengkapnya tentang cara menentukan Region default, lihat Mengonfigurasi Antarmuka Baris Perintah AWS.

Gunakan argumen berikut untuk instans Amazon EC2 dan lokal.

`--infrastructure-class`

(Wajib) Parameter ini harus disetel ke salah satu `ec2` atau`on-premises`, untuk menunjukkan apakah Anda mendaftarkan instans Amazon EC2 atau lokal.

`--stack-id`

(Wajib) ID tumpukan tempat instance akan didaftarkan.

> **Note**
>
> Untuk menemukan ID tumpukan, pada halaman Stack, pilih Pengaturan. ID tumpukan diberi label OpsWorks ID, dan merupakan GUID yang terlihat seperti. `ad21bce6-7623-47f1-bf9d-af2affad8907`

Argumen Login SSH

Gunakan argumen berikut untuk menentukan bagaimana `register` harus masuk ke instance.

`--local`

(Opsional) Gunakan argumen ini untuk mendaftarkan instance tempat Anda menjalankan perintah.

Dalam hal ini, `register` tidak perlu masuk ke instance.

`--ssh-private-key` dan **`--ssh-username`**

(Opsional) Gunakan argumen ini jika Anda mendaftarkan instance dari workstation terpisah dan ingin secara eksplisit menentukan nama pengguna atau file kunci pribadi.

- `--ssh-username`— Gunakan argumen ini untuk menentukan nama pengguna SSH.

  Jika Anda menghilangkan`--ssh-username`, `ssh` gunakan nama pengguna default.
- `--ssh-private-key`— Gunakan argumen ini untuk secara eksplisit menentukan file kunci pribadi.

  Jika Anda menghilangkan`--ssh-private-key`, `ssh` mencoba masuk menggunakan teknik otentikasi yang tidak memerlukan kata sandi, termasuk menggunakan kunci pribadi default. Jika tidak ada teknik yang didukung, `ssh` kueri untuk kata sandi Anda. Untuk informasi selengkapnya tentang cara `ssh` menangani otentikasi, lihat Protokol Otentikasi Secure Shell (SSH).

`--override-ssh`

(Opsional) Gunakan argumen ini jika Anda mendaftarkan instance dari workstation terpisah dan ingin menentukan string `ssh`perintah kustom. `register`Perintah menggunakan string perintah ini untuk masuk ke instance terdaftar.

Untuk informasi selengkapnya`ssh`, lihat SSH.

`--override-hostname`

(Opsional) Menentukan nama host untuk contoh, yang hanya digunakan oleh AWS OpsWorks Stacks. Nilai default adalah nama host instance.

`--debug`

(Opsional) Menyediakan informasi debugging jika proses pendaftaran gagal. Untuk informasi pemecahan masalah, lihat Pemecahan Masalah Pendaftaran Instans.

`--use-instance-profile`

(Opsional, tetapi sangat disarankan untuk EC2 instance Amazon) Memungkinkan `register` perintah menggunakan profil instance terlampir, alih-alih membuat pengguna IAM. Menambahkan parameter ini dapat membantu mencegah kesalahan yang terjadi jika Anda mencoba mengelola instance terdaftar ketika pengguna IAM secara tidak sengaja dihapus.

> **Important**
>
> Pengguna IAM yang dibuat selama proses pendaftaran diperlukan sepanjang masa pakai instance terdaftar. Menghapus pengguna menyebabkan agen AWS OpsWorks Stacks tidak dapat berkomunikasi dengan layanan. Untuk membantu mencegah masalah dalam mengelola instance terdaftar jika pengguna terhapus secara tidak sengaja, tambahkan `--use-instance-profile` parameter ke `register` perintah Anda untuk menggunakan profil instans bawaan instans sebagai gantinya. Menambahkan `--use-instance-profile` parameter juga mencegah kesalahan terjadi ketika Anda memutar kunci akses AWS akun setiap 90 hari (praktik terbaik yang disarankan), karena mencegah ketidakcocokan antara kunci akses yang tersedia untuk AWS OpsWorks agen dan pengguna yang diperlukan.

Target

(Bersyarat) Jika Anda menjalankan perintah ini dari workstation, nilai akhir dalam string perintah menentukan target pendaftaran dengan salah satu cara berikut.

- Alamat IP publik instans.
- Nama host instance.
- Untuk EC2 instans Amazon, ID instans.

AWS OpsWorks Stacks menggunakan ID instance untuk mendapatkan konfigurasi instance, termasuk alamat IP publik instans. Secara default, AWS OpsWorks Stacks menggunakan alamat ini untuk membangun string `ssh` perintah yang digunakannya untuk masuk ke instance. Jika Anda perlu terhubung ke alamat IP pribadi, Anda harus menggunakan `--override-ssh` untuk memberikan string perintah khusus. Sebagai contoh, lihat Mendaftarkan Instance Lokal dari Workstation.

> **Note**
>
> Jika Anda menentukan nama host, `ssh` tergantung pada server DNS untuk menyelesaikan nama ke instance tertentu. Jika Anda tidak yakin bahwa nama host unik, gunakan `ssh` untuk memverifikasi bahwa nama host diselesaikan ke instance yang benar.

Jika Anda menjalankan perintah ini dari instance yang akan didaftarkan, hilangkan pengenal instance dan gunakan argumennya`--local`.

Argumen berikut hanya untuk instance lokal.

`--override-public-ip`

(Opsional) AWS OpsWorks Stacks menampilkan alamat yang ditentukan sebagai alamat IP publik instans. Itu tidak mengubah alamat IP publik instance. Namun, jika pengguna menggunakan konsol untuk terhubung ke instance, seperti dengan memilih alamat di halaman Instans, AWS OpsWorks Stacks menggunakan alamat yang ditentukan. AWS OpsWorks Tumpukan secara otomatis menentukan nilai default argumen.

`--override-private-ip`

(Opsional) AWS OpsWorks Stacks menampilkan alamat yang ditentukan sebagai alamat IP pribadi instans. Itu tidak mengubah alamat IP pribadi instance. AWS OpsWorks Tumpukan secara otomatis menentukan nilai default argumen.

Contoh register Perintah

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Bagian ini berisi beberapa contoh string `register` perintah.

Mendaftarkan EC2 Instans Amazon dari Workstation

Contoh berikut mendaftarkan EC2 instance Amazon dari workstation. String perintah menggunakan kredensi default dan mengidentifikasi instance dengan ID instans EC2 Amazon-nya. Anda dapat menggunakan contoh untuk instance lokal dengan mengubah `ec2` ke. `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
```

Mendaftarkan Instance Lokal dari Workstation

Contoh berikut mendaftarkan instance lokal dari workstation terpisah. String perintah menggunakan kredensyal default dan log in ke instance dengan string `ssh` perintah yang

ditentukan. Jika instans Anda memerlukan kata sandi, `register` minta Anda. Anda dapat menggunakan contoh untuk EC2 instans Amazon dengan mengubah `on-premises` ke`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**
>
> Anda dapat menggunakan `--override-ssh` untuk menentukan string perintah SSH kustom. AWS OpsWorks Tumpukan kemudian menggunakan string yang ditentukan untuk masuk ke instance alih-alih membangun string perintah. Untuk contoh lain, lihatDaftarkan Instance Menggunakan String Perintah SSH Kustom.

Daftarkan Instance Menggunakan String Perintah SSH Kustom

Contoh berikut mendaftarkan instance lokal dari workstation, dan menggunakan `--override-ssh` argumen untuk menentukan perintah SSH kustom yang `register` digunakan untuk masuk ke instance. Contoh ini digunakan `sshpass` untuk masuk dengan nama pengguna dan kata sandi, tetapi Anda dapat menentukan string `ssh` perintah yang valid.

```
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"
```

Mendaftarkan Instance dengan Menjalankan **`register`** dari Instance

Contoh berikut menunjukkan cara mendaftarkan EC2 instance Amazon dengan menjalankan `register` dari instance itu sendiri. String perintah tergantung pada kredensi default untuk izinnya. Untuk menggunakan contoh untuk instance lokal, ubah `--infrastructure-class` ke`on-premises`.

```
aws opsworks register \
  --region us-west-2 \
  --infrastructure-class ec2 \
```

```
  --stack-id ad21bce6-7623-47f1-bf9d-af2affad8907 \
  --local
```

Daftarkan Instance dengan Alamat IP Pribadi

Secara default, `register` menggunakan alamat IP publik instans untuk masuk ke instance. Untuk mendaftarkan instance dengan alamat IP pribadi, seperti instance di subnet pribadi VPC, Anda harus menggunakan `--override-ssh` untuk menentukan string `ssh` perintah khusus.

```
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
```

Kebijakan Pendaftaran Instans

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

`AWSOpsWorksRegisterCLI_OnPremises`Kebijakan `AWSOpsWorksRegisterCLI_EC2` dan masing-masing memberikan izin yang benar untuk pendaftaran EC2 dan instans lokal. Anda menambahkan `AWSOpsWorksRegisterCLI_EC2` ke pengguna IAM Anda untuk mendaftarkan EC2 instance, tetapi tambahkan `AWSOpsWorksRegisterCLI_OnPremises` ke pengguna Anda untuk mendaftarkan instans lokal. Untuk menggunakan kebijakan ini, Anda harus menjalankan setidaknya versi 1.16.180 AWS CLI atau yang lebih baru.

`AWSOpsWorksRegisterCLI_EC2`Kebijakan

Tambahkan `AWSOpsWorksRegisterCLI_EC2` ke pengguna Anda untuk mendaftarkan EC2 instance. Anda harus menggunakan profil ini jika Anda berencana untuk mendaftar hanya EC2 instance. Saat Anda menggunakan kebijakan ini, izin diberikan oleh profil EC2 instans instans.

```
{
      "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": [
            "*"
          ]
        }
      ]
    }
```

AWSOpsWorksRegisterCLI_OnPremises Kebijakan

Tambahkan `AWSOpsWorksRegisterCLI_OnPremises` ke pengguna Anda untuk mendaftarkan instans lokal. Kebijakan ini mencakup izin IAM, seperti`AttachUserPolicy`, tetapi sumber daya tempat izin tersebut bekerja dibatasi.

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

(Usang) Kebijakan **AWSOpsWorksRegisterCLI**

> **Important**
>
> `AWSOpsWorksRegisterCLI`Kebijakan ini tidak digunakan lagi, dan tidak dapat digunakan untuk mendaftarkan instans baru. Ini hanya tersedia untuk kompatibilitas mundur pada instance yang telah terdaftar. `AWSOpsWorksRegisterCLI`Kebijakan ini mencakup banyak izin IAM termasuk`CreateUser`,`PutUserPolicy`, dan. `AddUserToGroup` Karena ini adalah izin tingkat administrator, Anda hanya boleh menetapkan `AWSOpsWorksRegisterCLI` kebijakan tersebut kepada pengguna administratif tepercaya.

Mengelola Instans Terdaftar

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Ketika Anda mendaftarkan sebuah instance, itu menjadi instance AWS OpsWorks Stacks, dan Anda dapat mengelolanya dengan cara yang hampir sama seperti instance yang dibuat dengan AWS OpsWorks Stacks. Ada dua perbedaan utama:

- Instance terdaftar tidak harus ditugaskan ke lapisan.
- Anda dapat membatalkan pendaftaran instance terdaftar dan mengembalikannya ke kontrol langsung Anda.

Setelah Anda mendaftarkan sebuah instance, itu dalam keadaan Terdaftar. AWS OpsWorks Stacks menyediakan fungsionalitas manajemen berikut untuk semua instance terdaftar:

- Pemeriksaan Kesehatan — AWS OpsWorks Stacks memantau agen untuk mengevaluasi apakah instance terus berfungsi.

  Jika instans gagal dalam pemeriksaan kesehatan, AWS OpsWorks Stacks akan menyembuhkan EC2 instans Amazon terdaftar secara otomatis dan mengubah status instans lokal yang terdaftar menjadi. `connection lost`
- CloudWatch pemantauan - CloudWatch pemantauan diaktifkan untuk contoh terdaftar.

  Anda dapat memantau metrik seperti pemanfaatan CPU dan memori yang tersedia dan secara opsional menerima pemberitahuan jika metrik melewati ambang batas yang ditentukan.
- Manajemen pengguna — AWS OpsWorks Stacks menyediakan cara sederhana untuk menentukan pengguna mana yang dapat mengakses instance dan operasi apa yang diizinkan untuk mereka lakukan. Untuk informasi selengkapnya, lihat Mengelola izin.
- Eksekusi resep — Anda dapat menggunakan perintah tumpukan Execute Recipes untuk mengeksekusi resep Chef pada instance.
- Pembaruan sistem operasi - Anda dapat menggunakan perintah tumpukan Update Dependencies untuk memperbarui sistem operasi instans.

Untuk memanfaatkan sepenuhnya fungsionalitas manajemen AWS OpsWorks Stacks, Anda dapat menetapkan instance ke lapisan. Untuk informasi selengkapnya, lihat Menetapkan Instance Terdaftar ke Layer.

Ada perbedaan antara cara AWS OpsWorks Stacks mengelola Amazon EC2 dan instans lokal.

EC2 Contoh Amazon

- Jika Anda menghentikan EC2 instans Amazon terdaftar, AWS OpsWorks Stacks menghentikan instans yang didukung penyimpanan instans dan menghentikan instans yang didukung Amazon EBS.

  Instance masih terdaftar dengan tumpukan dan ditugaskan ke lapisannya, sehingga Anda dapat memulai ulang jika diperlukan. Anda harus membatalkan pendaftaran instance terdaftar untuk menghapusnya dari tumpukan, baik secara eksplisit atau dengan menghapus instance, yang secara otomatis membatalkan pendaftarannya.
- Jika Anda memulai ulang EC2 instans Amazon terdaftar atau instans gagal dan disembuhkan secara otomatis, hasilnya sama dengan menghentikan dan memulai ulang instance dengan menggunakan Amazon. EC2 Perhatikan perbedaan ini:
  - Instans yang didukung penyimpanan instans — AWS OpsWorks Stacks memulai instance baru dengan AMI yang sama.

    Perhatikan bahwa AWS OpsWorks Stacks tidak memiliki pengetahuan tentang operasi apa pun yang Anda lakukan pada instance sebelum terdaftar, seperti menginstal paket perangkat lunak. Jika Anda ingin AWS OpsWorks Stacks menginstal paket atau melakukan tugas konfigurasi lainnya saat startup, Anda harus memberikan resep Chef khusus yang melakukan tugas yang diperlukan dan menetapkannya ke acara Pengaturan lapisan yang sesuai.
  - Instans yang didukung Amazon EBS — AWS OpsWorks Stacks memulai instance baru dengan AMI yang sama dan memasang kembali volume root, yang mengembalikan instance ke konfigurasi sebelumnya.
- Jika Anda membatalkan pendaftaran instans Amazon terdaftar, EC2 instans Amazon akan kembali menjadi instans Amazon biasa. EC2

Instans Lokal

- AWS OpsWorks Tumpukan tidak dapat menghentikan atau memulai instance lokal yang terdaftar.

Membatalkan penetapan instans lokal terdaftar akan memicu peristiwa Shutdown. Namun, acara itu hanya menjalankan resep Shutdown lapisan yang ditetapkan. Mereka melakukan tugas-tugas seperti mematikan layanan, tetapi tidak menghentikan instance.

- AWS OpsWorks Tumpukan tidak dapat melakukan autoheal instance lokal yang terdaftar jika gagal, tetapi instance ditandai sebagai koneksi terputus.
- Instans lokal tidak dapat menggunakan layanan Elastic Load Balancing, Amazon EBS, atau alamat IP Elastic.

Menetapkan Instance Terdaftar ke Layer

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Fitur ini hanya didukung untuk tumpukan Linux.

Setelah Anda mendaftarkan sebuah instance, Anda dapat menetapkannya ke satu atau lebih lapisan. Keuntungan menetapkan instance ke lapisan alih-alih membiarkannya tidak ditetapkan adalah Anda dapat menetapkan resep khusus ke peristiwa siklus hidup lapisan. AWS OpsWorks Tumpukan kemudian menjalankannya secara otomatis pada waktu yang tepat, setelah resep lapisan untuk acara itu.

- Anda dapat menetapkan instance terdaftar apa pun ke lapisan kustom. Lapisan kustom memiliki seperangkat resep minimal yang tidak menginstal paket apa pun, jadi mereka tidak boleh membuat konflik dengan konfigurasi instance yang ada.
- Anda dapat menetapkan instance lokal ke lapisan bawaan AWS OpsWorks Stacks.

Setiap lapisan bawaan menyertakan resep yang secara otomatis menginstal satu atau lebih paket. Misalnya, resep Java App Server Setup menginstal Apache dan Tomcat. Resep lapisan mungkin juga melakukan operasi lain seperti memulai ulang layanan dan menerapkan aplikasi. Sebelum menetapkan instance lokal ke lapisan bawaan, Anda harus memastikan bahwa resep lapisan tidak menimbulkan konflik apa pun, seperti mencoba menginstal versi server aplikasi yang berbeda dari yang saat ini ada di instance. Untuk informasi selengkapnya, silakan lihat Lapisan dan AWS OpsWorks Referensi Lapisan Tumpukan.

Untuk menetapkan instance terdaftar ke lapisan

1. Tambahkan layer yang ingin Anda gunakan ke tumpukan, jika Anda belum melakukannya.
2. Pilih Instans di panel navigasi dan kemudian pilih tetapkan di kolom Tindakan instance.
3. Pilih layer yang sesuai dan pilih Save.

Ketika Anda menetapkan sebuah instance ke layer AWS OpsWorks Stacks melakukan hal berikut.

- Menjalankan resep Setup layer.
- Menambahkan alamat IP Elastis terlampir atau volume Amazon EBS ke sumber daya tumpukan.

  Anda kemudian dapat menggunakan AWS OpsWorks Stacks untuk mengelola sumber daya ini. Untuk informasi selengkapnya, lihat Manajemen Sumber Daya.

Setelah selesai, instance berada dalam status online dan sepenuhnya dimasukkan ke dalam tumpukan. AWS OpsWorks Tumpukan kemudian menjalankan resep yang ditetapkan layer setiap kali peristiwa siklus hidup terjadi.

Membatalkan Penetapan Instance Terdaftar

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

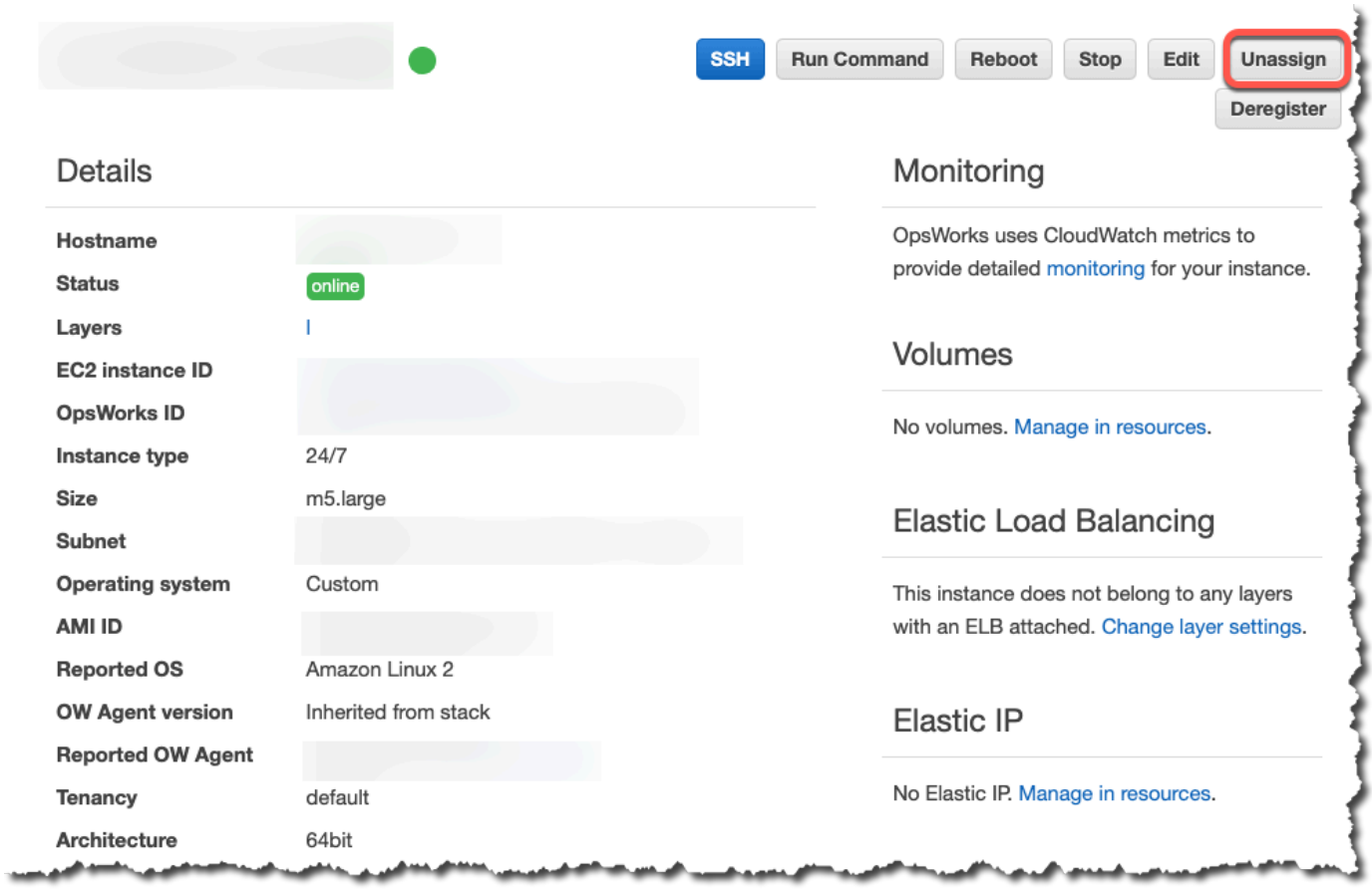
> **Note**
>
> Fitur ini hanya didukung untuk tumpukan Linux.

Anda dapat membatalkan penetapan instance terdaftar dari lapisannya dengan menggunakan AWS OpsWorks konsol, AWS CLI, atau operasi SDK.

Saat Anda membatalkan penetapan instance, AWS OpsWorks Stacks menjalankan resep Shutdown layer pada instance. Resep-resep ini melakukan tugas-tugas seperti mematikan layanan tetapi tidak menghentikan instance. Jika instance ditetapkan ke beberapa lapisan, unassign berlaku untuk setiap lapisan; Anda tidak dapat membatalkan penetapan instance dari subset lapisannya. Namun, instance masih terdaftar dengan tumpukan, dan Anda dapat menetapkannya ke lapisan lain jika diinginkan.

Untuk membatalkan penetapan instance terdaftar menggunakan konsol

1. Di panel navigasi, pilih Instans.
2. Pilih contoh yang ingin Anda batalkan.
3. Pada halaman Detail untuk contoh, pilih Unassign.



Untuk membatalkan penetapan instance terdaftar menggunakan AWS CLI

Jalankan aws opsworks unassign-instanceperintah untuk membatalkan penetapan instance terdaftar dari semua lapisan yang menggunakan instance.

```
aws opsworks unassign-instance --region region --instance-id instance-id
```

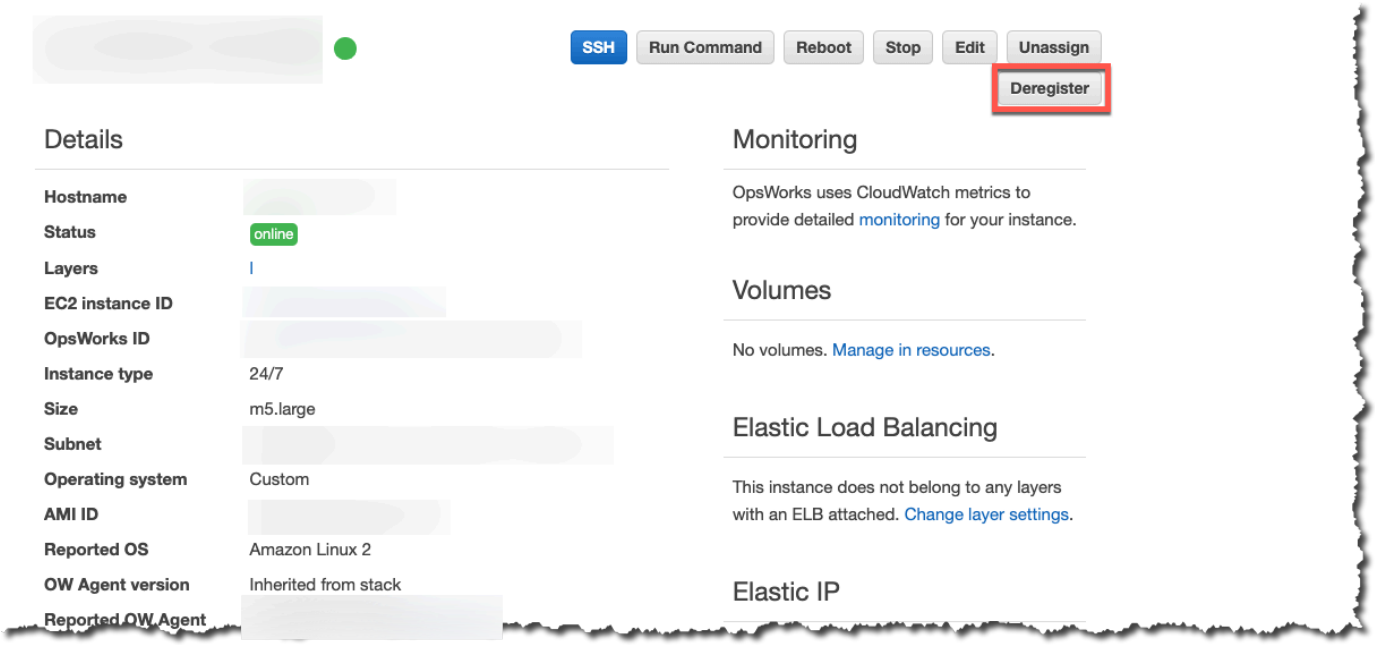
Menderegistrasi Instance Terdaftar

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat membatalkan pendaftaran instance menggunakan AWS OpsWorks konsol, AWS CLI, atau operasi SDK.

Untuk membatalkan pendaftaran instance menggunakan konsol

1. Di panel navigasi, pilih Instans.
2. Pilih contoh yang ingin Anda deregister.
3. Pada halaman Detail untuk contoh, pilih Deregister.



Untuk membatalkan pendaftaran instance menggunakan AWS CLI

Jalankan aws opsworks deregister-instanceperintah untuk membatalkan pendaftaran instance dari tumpukannya.

```
aws opsworks deregister-instance --region region --instance-id instance-id
```

Saat Anda membatalkan pendaftaran instance, AWS OpsWorks Stacks melakukan hal berikut:

- Menghapus instance dari tumpukan.
- Membatalkan penugasan instance dari setiap lapisan yang ditetapkan.
- Menutup dan menghapus instalan agen.
- Menderegister semua sumber daya terlampir (alamat IP Elastis dan volume Amazon EBS).

Prosedur ini mencakup sumber daya yang dilampirkan ke instance sebelum pendaftaran, dan sumber daya yang Anda gunakan AWS OpsWorks Stacks untuk dilampirkan ke instance saat itu adalah bagian dari tumpukan. Setelah deregistrasi, sumber daya tidak lagi menjadi bagian dari sumber daya tumpukan, tetapi mereka tetap melekat pada instance.

- Untuk instans lokal, hentikan tagihan.
- Menghapus semua tag yang OpsWorks ditambahkan ke instance.

Instans tetap dalam keadaan berjalan, tetapi berada di bawah kendali langsung Anda dan tidak lagi dikelola oleh AWS OpsWorks Stacks.

> **Note**
>
> Baik registrasi dan deregistering komputer atau instance didukung sepenuhnya hanya dalam tumpukan Linux. Untuk tumpukan Windows, instance deregistering diperbolehkan, tetapi tidak menghapus instalan agen dari instance. OpsWorks Deregistrasi tidak menghapus semua file yang diubah, dan tidak sepenuhnya kembali ke salinan cadangan file tertentu. Daftar ini berlaku untuk tumpukan Chef 11.10 dan Chef 12; perbedaan antara kedua versi dicatat di sini.
>
> - `/etc/hosts`didukung ke`/var/lib/aws/opsworks/local-mode-cache/backup/etc/`, tetapi tidak dipulihkan.
> - Entri tetap untuk `aws` dan `opsworks` dalam file passwd, grup, dan bayangan, dll.
> - `/etc/sudoers`berisi referensi ke direktori AWS OpsWorks Stacks.
> - File-file berikut aman untuk ditinggalkan; jangka panjang, pertimbangkan untuk menghapus. `/var/lib/aws/opsworks`
>   - `/var/log/aws/opsworks`tetap pada instance di tumpukan Chef 11.10.
>   - `/var/lib/aws/opsworks`tetap ada di tumpukan Chef 11.10 dan Chef 12.
>   - `/var/chef`tetap pada instance di tumpukan Chef 12.
> - File lain yang tertinggal:
>   - `/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`

Siklus Hidup Instans Terdaftar

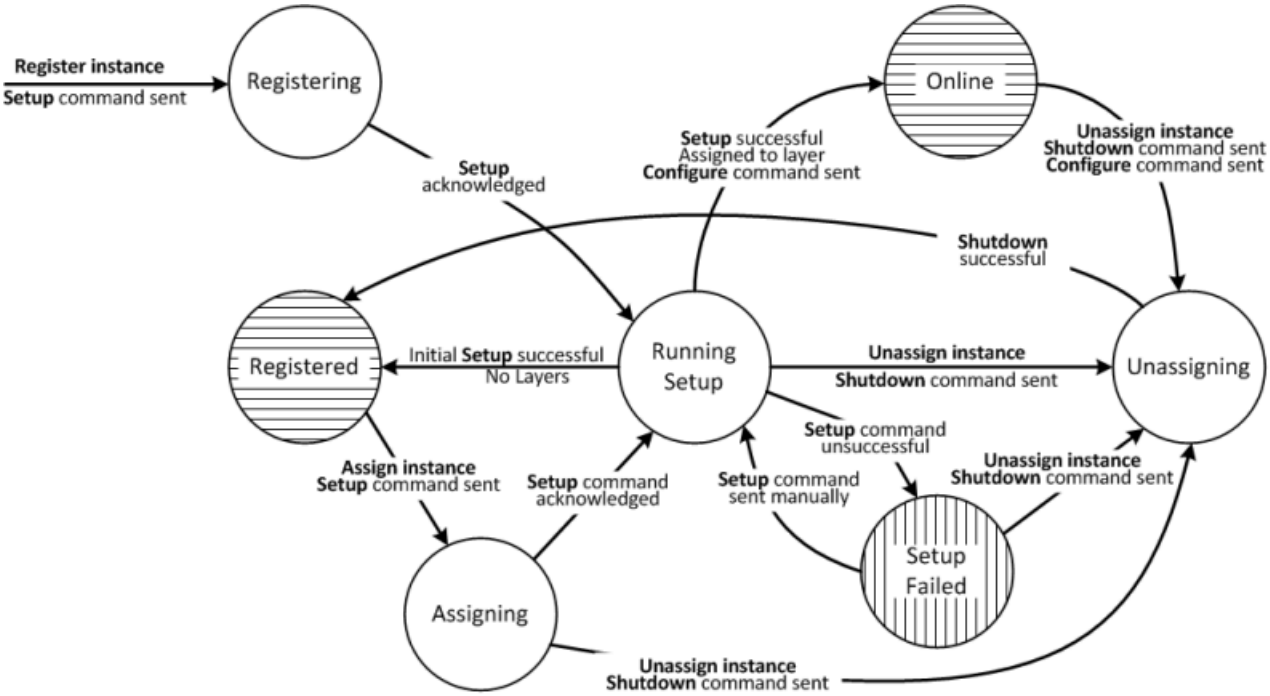
Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Fitur ini hanya didukung untuk tumpukan Linux.

Siklus hidup instance terdaftar dimulai setelah agen diinstal dan dijalankan. Pada saat itu, ia mengarahkan AWS OpsWorks Stacks untuk mendaftarkan instance dengan tumpukan. Diagram status berikut merangkum elemen siklus hidup utama.



Setiap negara sesuai dengan status instance. Tepi mewakili salah satu perintah AWS OpsWorks Stacks berikut. Rinciannya dibahas di bagian berikut.

- Pengaturan - Perintah ini sesuai dengan peristiwa siklus hidup Pengaturan dan menjalankan resep Pengaturan instans.
- Konfigurasi - Perintah ini sesuai dengan peristiwa Configure lifecycle.

  AWS OpsWorks Tumpukan memicu peristiwa ini pada setiap instance dalam tumpukan ketika sebuah instance memasuki atau meninggalkan status online. Instance menjalankan resep Configure mereka, yang membuat perubahan apa pun yang diperlukan untuk mengakomodasi instance baru.
- Shutdown - Perintah ini sesuai dengan peristiwa siklus hidup Shutdown, yang menjalankan resep Shutdown instance.

  Resep-resep ini melakukan tugas-tugas seperti mematikan layanan, tetapi mereka tidak menghentikan instance.
- Deregister — Perintah ini membatalkan pendaftaran instance dan tidak sesuai dengan peristiwa siklus hidup.

> **Note**
>
> Untuk mempermudah diagram tidak menunjukkan status Deregistering dan Deleted. Anda dapat membatalkan pendaftaran instance dari salah satu status dalam diagram, yang mengirimkan perintah Deregister ke instance dan memindahkannya ke status Deregistering.
>
> - Jika Anda membatalkan pendaftaran instance online, AWS OpsWorks Stacks mengirimkan perintah Configure ke instance yang tersisa di tumpukan untuk memberi tahu mereka bahwa instance sedang offline.
> - Setelah perintah Deregister diakui, instance masih berjalan, tetapi dalam status Dihapus dan tidak lagi menjadi bagian dari tumpukan. Jika Anda ingin memasukkan instance ke dalam tumpukan lagi, Anda harus mendaftarkannya kembali.

Topik

- Mendaftar
- Menjalankan Pengaturan
- Terdaftar
- Menugaskan
- Online
- Penyiapan Gagal
- Membatalkan penugasan
- Perubahan Konfigurasi Pengaturan Awal

Mendaftar

Setelah agen mengirimkan permintaan pendaftaran, AWS OpsWorks Stacks memulai siklus hidup instance dengan mengirimkan perintah Setup ke instance, meletakkannya di status Registrasi. Setelah instance mengakui perintah Setup, ia pindah ke Menjalankan Pengaturan state.

Menjalankan Pengaturan

Status Running Setup menjalankan resep Pengaturan instans. Pengaturan berfungsi tergantung pada keadaan sebelumnya.

> **Note**
>
> Jika Anda membatalkan penetapan instance saat berada dalam status Running Setup, AWS OpsWorks Stacks mengirimkan perintah Shutdown, yang menjalankan resep Shutdown instance tetapi tidak menghentikan instance. Instance pindah ke Membatalkan penugasan negara.

Topik

- Mendaftar
- Menugaskan
- Penyiapan Gagal

Mendaftar

Selama proses Pendaftaran, penyiapan membuat instance AWS OpsWorks Stacks untuk mewakili instance terdaftar di tumpukan, dan menjalankan satu set resep Setup inti pada instance.

Salah satu perubahan utama yang dilakukan oleh penyiapan awal adalah menimpa file host instance. Dengan mendaftarkan instance, Anda telah menyerahkan manajemen pengguna ke AWS OpsWorks Stacks, yang harus memiliki file host sendiri untuk mengontrol izin login SSH. Pengaturan awal juga membuat atau memodifikasi sejumlah file dan, pada sistem Ubuntu, memodifikasi sumber paket dan menginstal satu set paket. Untuk detailnya, lihat Perubahan Konfigurasi Pengaturan Awal.

Selama pendaftaran, proses memanggil IAM `AttachUserPolicy` yang merupakan bagian dari izin yang dilampirkan ke pengguna IAM yang Anda buat sebagai prasyarat. Jika `AttachUserPolicy` tidak ada (kemungkinan besar karena Anda menjalankan rilis AWS CLI yang lebih lama), prosesnya kembali ke panggilan. `PutUserPolicy`

> **Note**
>
> Untuk konsistensi, AWS OpsWorks Stacks menjalankan setiap resep Pengaturan inti. Namun, beberapa dari mereka melakukan beberapa atau semua tugas mereka hanya jika sebuah instance telah ditugaskan ke setidaknya satu lapisan, sehingga mereka tidak selalu mempengaruhi pengaturan awal.

- Jika penyiapan berhasil, instance pindah ke Terdaftar status.

- Jika penyiapan tidak berhasil, instance pindah ke Penyiapan Gagal status.

Menugaskan

Instance memiliki setidaknya satu lapisan yang ditetapkan. AWS OpsWorks Stacks menjalankan resep Pengaturan setiap lapisan, termasuk resep khusus apa pun yang telah Anda tetapkan ke acara Pengaturan lapisan.

- Jika penyiapan berhasil, instance pindah ke status Online dan AWS OpsWorks Stacks memicu peristiwa siklus hidup Konfigurasi pada setiap instance di tumpukan untuk memberi tahu mereka tentang instance baru.
- Jika penyiapan tidak berhasil, instance pindah ke status Setup Failed.

> **Note**
>
> Proses penyiapan ini menjalankan resep inti untuk kedua kalinya. Namun, resep Chef sangat idempoten, sehingga tidak mengulangi tugas apa pun yang telah dilakukan.

Penyiapan Gagal

Jika proses penyiapan untuk instance dalam Menugaskan status gagal, Anda dapat mencoba lagi dengan menggunakan perintah Setup stack untuk menjalankan kembali resep Pengaturan instans secara manual.

- Jika penyiapan berhasil, instance yang ditetapkan pindah ke Online status dan AWS OpsWorks Stacks memicu peristiwa siklus hidup Konfigurasi pada setiap instance di tumpukan untuk memberi tahu mereka tentang instance baru.
- Jika upaya penyiapan tidak berhasil, instance akan kembali ke status Setup Failed.

Terdaftar

Instance dalam status Terdaftar adalah bagian dari tumpukan dan dikelola oleh AWS OpsWorks Stacks tetapi tidak ditetapkan ke lapisan. Mereka dapat tetap dalam keadaan ini tanpa batas waktu.

Jika Anda menetapkan instance ke satu atau beberapa layer, AWS OpsWorks Stacks mengirimkan perintah Setup ke instance dan pindah ke state. Menugaskan

Menugaskan

Setelah instance mengakui perintah Setup, ia pindah ke Menjalankan Pengaturan state.

Jika Anda membatalkan penetapan instance saat berada dalam status Menetapkan, AWS OpsWorks Stacks menghentikan proses penyiapan dan mengirimkan perintah Shutdown. Instance pindah ke Membatalkan penugasan negara.

Online

Instance sekarang menjadi anggota dari setidaknya satu lapisan dan diperlakukan seperti instance AWS OpsWorks Stacks biasa. Itu bisa tetap dalam keadaan ini tanpa batas waktu.

Jika Anda membatalkan penetapan instance saat berada dalam status Online, AWS OpsWorks Stacks mengirimkan perintah Shutdown ke instance dan perintah Configure ke instance stack lainnya. Instance pindah ke Membatalkan penugasan negara.

Penyiapan Gagal

Perintah Setup telah gagal.

- Anda dapat mencoba lagi dengan menjalankan perintah Setup stack.

  Instance kembali ke Menjalankan Pengaturan negara.
- Jika Anda membatalkan penetapan instance, AWS OpsWorks Stacks akan mengirimkan perintah Shutdown ke instance.

  Instance pindah ke Membatalkan penugasan negara.

Membatalkan penugasan

Setelah perintah Shutdown selesai, instance tidak lagi ditetapkan ke lapisan mana pun dan kembali ke status. Terdaftar

> **Note**
>
> Jika sebuah instance ditetapkan ke beberapa lapisan, pembatalan tugas berlaku untuk setiap lapisan; Anda tidak dapat membatalkan penetapan subset dari lapisan yang ditetapkan. Jika Anda menginginkan kumpulan lapisan yang ditetapkan yang berbeda, batalkan penetapan instance dan kemudian tetapkan kembali lapisan yang diinginkan.

Perubahan Konfigurasi Pengaturan Awal

Pengaturan awal membuat atau memodifikasi file dan direktori berikut pada semua instance terdaftar.

File yang Dibuat

```
/etc/apt/apt.conf.d/99-no-pipelining
/etc/aws/
/etc/init.d/opsworks-agent
/etc/motd
/etc/motd.opsworks-static
/etc/sudoers.d/opsworks
/etc/sudoers.d/opsworks-agent
/etc/sysctl.d/70-opsworks-defaults.conf
/opt/aws/opsworks/
/usr/sbin/opsworks-agent-cli
/var/lib/aws/
/var/log/aws/
/vol/
```

File yang Dimodifikasi

```
/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
```

Penyiapan awal juga membuat file swap di instans EC2 mikro Amazon.

Pengaturan awal membuat perubahan berikut pada sistem Ubuntu.

Sumber Package

Pengaturan awal mengubah sumber paket menjadi berikut ini.

- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`

Untuk: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} main universe`

- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`

  Untuk: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates main universe`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`

  Untuk: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security main universe`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`

  Untuk: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name}-updates multiverse`
- `deb http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Untuk: `deb-src http://archive.ubuntu.com/ubuntu ${code_name}-security multiverse`
- `deb http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`

  Untuk: `deb-src http://archive.ubuntu.com/ubuntu/ ${code_name} multiverse`
- `deb http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

  Untuk: `deb-src http://security.ubuntu.com/ubuntu ${code_name}-security multiverse`

Paket

Penyiapan awal menghapus instalasi `landscape` dan menginstal paket-paket berikut.

`autofs`	`libicu-dev`	`libopenssl-ruby`
`libssl-dev`	`libxml2-dev`	`libxslt-dev`
`libyaml-dev`	`monit`	`ntpd`

procps	ruby	ruby-dev
rubygems	screen	sqlite
vim	xfs	

Mengedit Konfigurasi Instance

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat mengedit konfigurasi instans, termasuk instans Amazon Elastic Compute Cloud EC2 (Amazon) terdaftar, dengan batasan berikut:

- Instans harus berada dalam status berhenti.

  Meskipun Anda tidak dapat memodifikasi properti instans online, Anda dapat mengubah beberapa aspek konfigurasinya dengan mengedit lapisan instans. Untuk informasi selengkapnya, lihat Mengedit Konfigurasi OpsWorks Layer.
- Beberapa pengaturan, seperti Availability Zone dan Scaling Type, ditentukan saat Anda membuat instance dan tidak dapat diubah nanti.
- Beberapa pengaturan dapat dimodifikasi hanya untuk instans yang didukung toko saja, bukan untuk instans yang didukung Amazon Elastic Block Store.

  Misalnya, Anda dapat mengubah sistem operasi instans yang didukung penyimpanan instans. Instans yang didukung Amazon EBS harus menggunakan sistem operasi yang Anda tentukan saat membuat instance. Untuk informasi selengkapnya tentang penyimpanan instans, lihat Penyimpanan.
- Secara default, instance mewarisi pengaturan versi agen stack.

Anda dapat menggunakan Versi OpsWorks Agen untuk mengganti setelan versi agen stack dan menentukan versi agen tertentu untuk sebuah instance. Jika Anda menentukan versi agen instans, AWS OpsWorks Stacks tidak secara otomatis memperbarui agen saat versi baru tersedia, meskipun pengaturan versi agen tumpukan adalah Pembaruan otomatis. Anda harus memperbarui versi agen instans secara manual dengan mengedit konfigurasi instans. AWS OpsWorks Tumpukan kemudian menginstal versi agen yang ditentukan pada instance.

> **Note**
>
> Anda tidak dapat mengedit konfigurasi instans lokal yang terdaftar.

Untuk mengedit konfigurasi instans

1. Hentikan instance, jika belum dihentikan.
2. Pada halaman Instans, klik nama instance untuk menampilkan halaman Detail.
3. Klik Edit untuk menampilkan halaman edit.
4. Edit konfigurasi instans, yang sesuai.

Untuk deskripsi nama Host, Ukuran, kunci SSH, dan pengaturan sistem operasi, lihatMenambahkan Instance ke Layer. Pengaturan Layers memungkinkan Anda menambah atau menghapus layer. Layer instance saat ini muncul mengikuti daftar layer.

- Untuk menambahkan layer lain, pilih dari daftar.
- Untuk menghapus instance dari salah satu lapisannya, klik x dengan layer yang sesuai.

  Sebuah instance harus menjadi anggota dari setidaknya satu lapisan, sehingga Anda tidak dapat menghapus lapisan terakhir.

Saat memulai ulang instans, AWS OpsWorks Stacks memulai EC2 instans Amazon baru dengan konfigurasi yang diperbarui.

Menghapus Instans AWS OpsWorks Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menggunakan AWS OpsWorks Stacks untuk menghentikan instans, termasuk EC2 instans Amazon terdaftar. Melakukannya menghentikan EC2 instance, tetapi instance tetap berada di tumpukan. Anda dapat memulai ulang dengan mengklik mulai di kolom Tindakan instans. Jika Anda tidak lagi memerlukan instance dan ingin menghapusnya dari tumpukan, Anda dapat menghapusnya, yang menghapus instance dari tumpukan dan mengakhiri EC2 instance Amazon terkait. Menghapus instance juga akan menghapus semua log atau data terkait, dan volume Amazon Elastic Block Store (EBS) Elastic Block Store (EBS) pada instans.

> **Important**
>
> Topik ini hanya berlaku untuk EC2 instans Amazon yang dikelola oleh AWS OpsWorks Stacks. Untuk informasi selengkapnya tentang cara menghapus instance yang dikelola oleh EC2 konsol Amazon atau API, lihat Menghentikan Instans Anda.

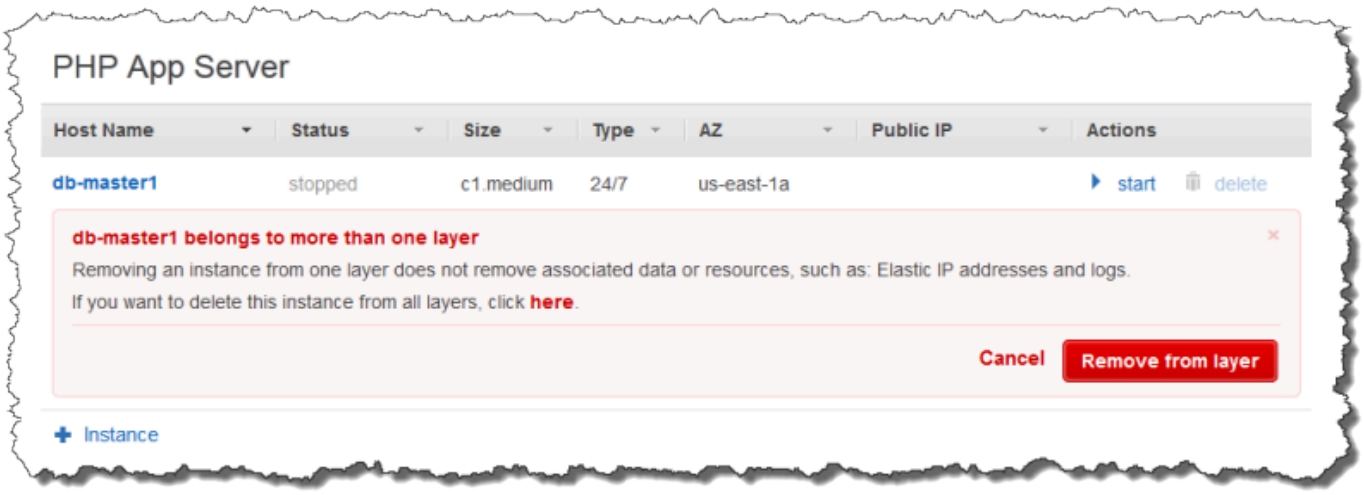
> **Note**
>
> Anda tidak dapat menggunakan AWS OpsWorks Stacks untuk menghapus instans lokal yang terdaftar.

Jika sebuah instance milik beberapa lapisan, Anda dapat menghapus instance dari tumpukan atau hanya menghapus lapisan tertentu. Anda juga dapat menghapus lapisan dari instance dengan mengedit konfigurasi instance, seperti yang dijelaskan dalamMengedit Konfigurasi Instance.

> **Important**
>
> Anda harus menghapus instance AWS OpsWorks Stacks hanya dengan menggunakan konsol AWS OpsWorks Stacks atau API. Secara khusus, Anda tidak boleh menghapus instance AWS OpsWorks Stacks dengan menggunakan EC2 konsol Amazon atau API karena EC2 tindakan Amazon tidak disinkronkan secara otomatis dengan Stacks. AWS OpsWorks Misalnya, jika penyembuhan otomatis diaktifkan dan Anda menghentikan instance dengan menggunakan EC2 konsol Amazon, AWS OpsWorks Stacks memperlakukan instance yang dihentikan sebagai instance gagal dan meluncurkan instance EC2 Amazon lain untuk menggantikannya. Untuk informasi selengkapnya, lihat Menggunakan Penyembuhan Otomatis.

Untuk menghapus sebuah instance

1. Pada halaman Instances, cari instance di bawah layer yang sesuai. Jika instance sedang berjalan, klik stop di kolom Actions.
2. Setelah status berubah menjadi berhenti, klik hapus. Jika instance adalah anggota lebih dari satu layer, layer AWS OpsWorks Stacks menampilkan bagian berikut.



- Untuk menghapus instance hanya dari lapisan yang dipilih, klik Hapus dari lapisan.

  Instance tetap menjadi anggota lapisan lainnya dan dapat dimulai ulang.
- Untuk menghapus instance dari semua lapisannya, yang menghapusnya dari tumpukan, klik di sini.

3. Jika Anda memilih untuk sepenuhnya menghapus instance dari tumpukan, atau jika instance adalah anggota hanya satu lapisan, AWS OpsWorks Stacks meminta Anda untuk mengonfirmasi penghapusan.

   Pilih Hapus untuk mengonfirmasi. Selain menghapus instance dari tumpukan, tindakan ini menghapus log atau data terkait, dan volume root yang dilampirkan ke instance. Untuk menghapus semua volume instans, pilih Hapus volume EBS instans (snapshot tidak akan dihapus) sebelum Anda memilih Hapus.

Menggunakan SSH untuk Masuk ke Instance Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat masuk ke instance Linux online Anda dengan SSH menggunakan MindTerm klien bawaan, atau klien pihak ketiga, seperti PuTTY. SSH biasanya bergantung pada key pair RSA untuk otentikasi. Anda menginstal kunci publik pada instance dan memberikan kunci pribadi yang sesuai ke klien SSH. AWS OpsWorks Stacks menangani pemasangan kunci publik pada instance tumpukan Anda untuk Anda, sebagai berikut.

- Amazon Elastic Compute Cloud (Amazon EC2) key pair — Jika wilayah stack memiliki satu atau beberapa pasangan EC2 kunci Amazon, Anda dapat menentukan key pair SSH default untuk stack.

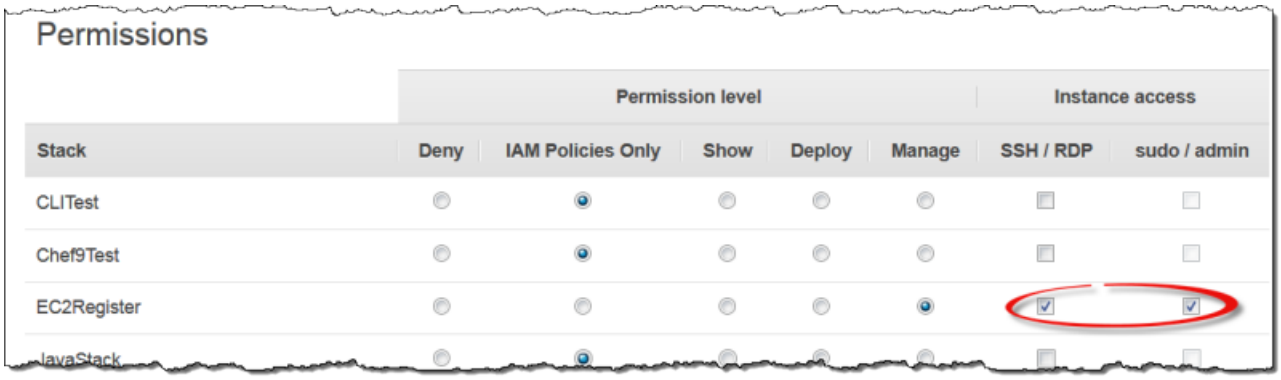
  Anda dapat secara opsional mengganti key pair default dan menentukan pasangan yang berbeda saat membuat instance. Dalam kedua kasus tersebut, AWS OpsWorks Stacks menginstal kunci publik pasangan kunci yang ditentukan pada instance. Untuk informasi selengkapnya tentang cara membuat pasangan EC2 kunci Amazon, lihat Pasangan EC2 Kunci Amazon.
- Personal key pair — Setiap pengguna dapat mendaftarkan personal key pair dengan AWS OpsWorks Stacks.

Pengguna atau administrator mendaftarkan kunci publik dengan AWS OpsWorks Stacks, dan pengguna menyimpan kunci pribadi secara lokal. Saat menyetel izin untuk tumpukan, administrator menentukan pengguna mana yang harus memiliki akses SSH ke instance tumpukan. AWS OpsWorks Stacks secara otomatis membuat pengguna sistem pada instance tumpukan untuk setiap pengguna yang berwenang dan menginstal kunci publik pribadi pengguna.

Seorang pengguna harus memiliki otorisasi SSH untuk menggunakan klien MindTerm SSH atau menggunakan personal key pair mereka untuk masuk ke instance stack.

Untuk mengotorisasi SSH untuk pengguna

1. Di panel navigasi AWS OpsWorks Stacks, klik Izin.
2. Pilih SSH/RDP untuk pengguna IAM yang diinginkan untuk memberikan izin yang diperlukan. Jika Anda ingin mengizinkan pengguna menggunakan untuk meningkatkan hak istimewa—misalnya, **sudo** untuk menjalankan perintah CLI agen—pilih sudo/admin juga.



Untuk informasi selengkapnya tentang cara menggunakan AWS OpsWorks Stacks untuk mengelola akses SSH, lihat. Mengelola Akses SSH

Topik

- Menggunakan Built-in MindTerm SSH Client
- Menggunakan Klien SSH Pihak Ketiga

Menggunakan Built-in MindTerm SSH Client

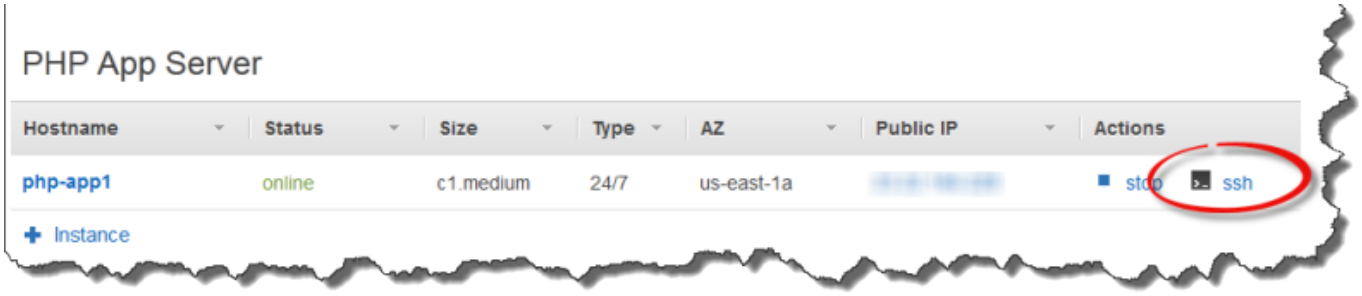
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Cara termudah untuk masuk ke instance Linux adalah dengan menggunakan klien MindTerm SSH bawaan. Setiap instance online menyertakan tindakan ssh yang dapat Anda gunakan untuk meluncurkan MindTerm klien.

> **Note**
>
> Anda harus mengaktifkan Java di browser Anda untuk menggunakan MindTerm klien.

Untuk masuk dengan MindTerm klien

1. Jika Anda belum melakukannya, otorisasi akses SSH untuk pengguna IAM yang akan terhubung ke instance, seperti yang dijelaskan di bagian sebelumnya.
2. Masuk sebagai pengguna.
3. Pada halaman Instances, pilih ssh di kolom Actions untuk instance yang sesuai.



4. Untuk kunci Privat, berikan jalur ke kunci pribadi pribadi pengguna atau kunci EC2 pribadi Amazon, tergantung pada kunci publik mana yang telah Anda instal pada instance.
5. Pilih Launch Mindterm dan gunakan jendela terminal untuk menjalankan perintah pada instance.

Menggunakan Klien SSH Pihak Ketiga

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda juga dapat menggunakan klien SSH pihak ketiga, seperti PuTTY, untuk terhubung ke instance Linux.

Untuk menggunakan klien SSH pihak ketiga

1. Pastikan AWS OpsWorks Stacks telah menginstal kunci EC2 publik Amazon atau kunci publik pribadi pengguna IAM pada instance, seperti yang dibahas sebelumnya.
2. Dapatkan nama DNS publik instans atau alamat IP publik dari halaman detailnya.
3. Berikan klien dengan nama host instance, yang bergantung pada sistem operasi, sebagai berikut:

   - Amazon Linux dan Red Hat Enterprise Linux (RHEL) - `ec2-user@`*`DNSName/Address`*`.`
   - Ubuntu —`ubuntu@`*`DNSName/Address`*.

   Ganti *`DNSName/Address`* dengan nama DNS publik atau alamat IP dari langkah sebelumnya.
4. Berikan klien dengan kunci pribadi yang sesuai dengan kunci publik yang diinstal. Anda dapat menggunakan kunci EC2 pribadi Amazon atau kunci pribadi pribadi pengguna IAM, tergantung pada kunci publik mana yang telah diinstal pada instance.

Menggunakan RDP untuk Masuk ke Instans Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menggunakan protokol desktop jarak jauh Windows (RDP) untuk masuk ke instance Windows online, sebagai berikut:

- Instance harus memiliki grup keamanan dengan aturan masuk yang memungkinkan akses RDP.

  Untuk informasi selengkapnya tentang bekerja dengan grup keamanan, lihatMenggunakan Grup Keamanan.
- Pengguna biasa — AWS OpsWorks Stacks menyediakan pengguna biasa yang berwenang dengan kata sandi RDP yang berlaku untuk jangka waktu terbatas, yang dapat berkisar dari 30 menit hingga 12 jam.

  Selain diberi wewenang, pengguna harus memiliki setidaknya tingkat izin Tampilkan atau kebijakan terlampir AWS Identity and Access Management (IAM) mereka harus mengizinkan `opsworks:GrantAccess` tindakan tersebut.
- Administrator — Anda dapat menggunakan kata sandi Administrator untuk masuk dalam waktu yang tidak terbatas.

  Seperti yang dijelaskan nanti, jika Anda telah menetapkan key pair Amazon Elastic Compute Cloud (Amazon EC2) untuk instance, Anda dapat menggunakannya untuk mengambil kata sandi Administrator.

> **Note**
>
> Topik ini menjelaskan cara menggunakan klien Windows Remote Desktop Connection untuk masuk dari workstation Windows. Anda juga dapat menggunakan salah satu klien RDP yang tersedia untuk Linux atau OS X, tetapi prosedurnya mungkin agak berbeda. Untuk informasi selengkapnya tentang klien RDP yang kompatibel dengan Microsoft Windows Server 2012 R2, lihat Klien Microsoft Remote Desktop.

Topik

- Menyediakan Grup Keamanan yang Memungkinkan Akses RDP

- Masuk Sebagai Pengguna Biasa
- Masuk Sebagai Administrator

Menyediakan Grup Keamanan yang Memungkinkan Akses RDP

Sebelum Anda dapat menggunakan RDP untuk masuk ke instance Windows, aturan masuk grup keamanan instans harus mengizinkan koneksi RDP. Saat Anda membuat tumpukan pertama di suatu wilayah, AWS OpsWorks Stacks membuat satu set grup keamanan. Mereka termasuk satu bernama sesuatu seperti`AWS-OpsWorks-RDP-Server`, yang AWS OpsWorks Stacks melampirkan ke semua instance Windows untuk memungkinkan akses RDP. Namun, secara default, grup keamanan ini tidak memiliki aturan apa pun, jadi Anda harus menambahkan aturan masuk untuk mengizinkan akses RDP ke instans Anda.

Untuk memungkinkan akses RDP

1. Buka EC2konsol Amazon, atur ke wilayah tumpukan, dan pilih Grup Keamanan dari panel navigasi.
2. Pilih AWS- OpsWorks -RDP-Server, pilih tab Inbound, dan pilih Edit.
3. Pilih Tambahkan Aturan dan tentukan pengaturan berikut:
   - Jenis - RDP
   - Sumber — Alamat IP sumber yang diizinkan.

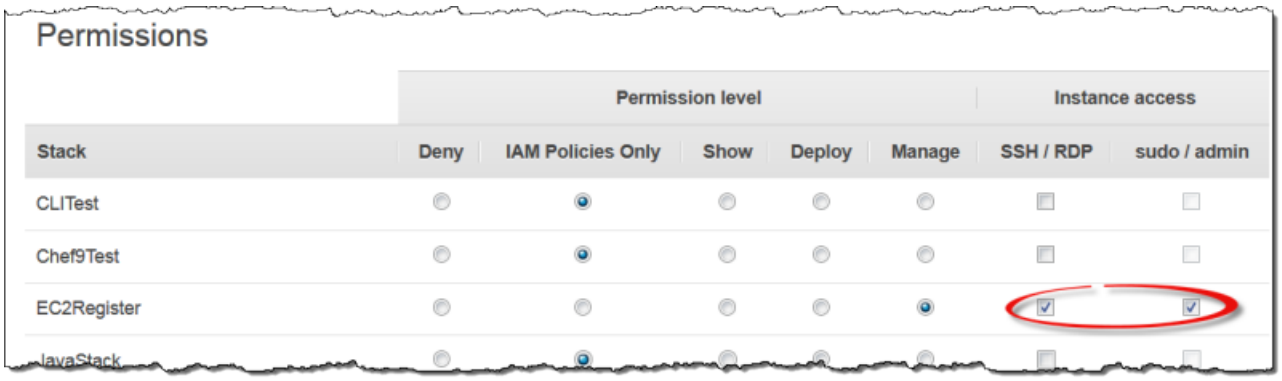
     Anda biasanya mengizinkan permintaan RDP masuk dari alamat IP Anda atau rentang alamat IP tertentu (biasanya rentang alamat IP perusahaan Anda).

Masuk Sebagai Pengguna Biasa

Pengguna yang berwenang dapat masuk ke instance menggunakan kata sandi sementara, yang disediakan oleh AWS OpsWorks Stacks.

Untuk mengotorisasi RDP untuk pengguna;

1. Di panel navigasi AWS OpsWorks Stacks, klik Izin.
2. Pilih kotak centang SSH/RDP untuk pengguna yang diinginkan untuk memberikan izin yang diperlukan. Jika Anda ingin pengguna memiliki izin administrator, Anda juga harus memilih sudo/admin.



Pengguna yang berwenang dapat masuk ke salah satu instance online stack, sebagai berikut.

Untuk masuk sebagai pengguna IAM biasa

1. Masuk sebagai pengguna IAM.
2. Pada halaman Instances, pilih rdp di kolom Actions untuk instance yang sesuai.
3. Tentukan panjang sesi, yang dapat bervariasi dari 30 menit hingga 12 jam, dan pilih Hasilkan Kata Sandi. Kata sandi hanya akan berlaku untuk durasi sesi yang ditentukan.
4. Catat nama DNS publik, nama pengguna, dan nilai kata sandi, lalu pilih Akui dan tutup.
5. Buka klien Windows Remote Desktop Connection, pilih Tampilkan Opsi, dan berikan yang berikut ini dari informasi yang Anda rekam di Langkah 4:

   - Komputer — Nama DNS publik instans.

     Anda juga dapat menggunakan alamat IP publik, jika Anda mau. Pilih Instans dan salin alamat dari kolom IP Publik instans.

   - Nama pengguna - Nama pengguna.
6. Saat klien meminta kredensil Anda, masukkan kata sandi yang Anda simpan di Langkah 4.

> **Note**
>
> AWS OpsWorks Tumpukan menghasilkan kata sandi pengguna hanya untuk instance online. Jika Anda memulai sebuah instance dan, misalnya, salah satu resep Pengaturan kustom Anda gagal, instance akan berakhir di `setup_failed` status. Meskipun instance tidak online sejauh menyangkut AWS OpsWorks Stacks, EC2 instance sedang berjalan dan seringkali berguna untuk masuk untuk memecahkan masalah. AWS OpsWorks Stacks tidak akan menghasilkan kata sandi untuk Anda dalam kasus ini, tetapi jika Anda telah

menetapkan key pair SSH ke instance, Anda dapat menggunakan EC2 konsol atau CLI untuk mengambil kata sandi Administrator instance dan masuk sebagai Administrator. Untuk informasi selengkapnya, lihat bagian berikut.

Masuk Sebagai Administrator

Anda dapat masuk ke instance sebagai Administrator dengan menggunakan kata sandi yang sesuai. Jika Anda telah menetapkan EC2 key pair ke instance, Amazon EC2 menggunakannya untuk secara otomatis membuat dan mengenkripsi kata sandi Administrator saat instance dimulai. Anda kemudian dapat menggunakan kunci pribadi pasangan kunci dengan EC2 konsol, API, atau CLI untuk mengambil dan mendekripsi kata sandi.

> **Note**
>
> Anda tidak dapat menggunakan key pair SSH pribadi untuk mengambil kata sandi Administrator; Anda harus menggunakan EC2 key pair.

Berikut ini menjelaskan cara menggunakan EC2 konsol untuk mengambil kata sandi Administrator dan masuk ke sebuah instance. Jika Anda lebih suka alat baris perintah, Anda juga dapat menggunakan `get-password-data` perintah AWS CLI untuk mengambil kata sandi.

Untuk masuk sebagai Administrator

1. Pastikan Anda telah menentukan EC2 key pair untuk instance tersebut. Anda dapat menentukan key pair default untuk semua instance stack ketika Anda membuat stack, atau Anda dapat menentukan key pair untuk instance tertentu ketika Anda membuat instance.
2. Buka EC2 konsol, atur ke wilayah tumpukan, dan pilih Instans dari panel navigasi.
3. Pilih instans, pilih Connect, dan pilih Get Password.
4. Berikan jalur ke kunci pribadi pasangan EC2 kunci di workstation Anda, dan pilih Dekripsi Kata Sandi. Salin kata sandi yang didekripsi untuk digunakan nanti.
5. Buka klien Windows Remote Desktop Connection, pilih Tampilkan Opsi, dan berikan informasi berikut:

   - Komputer — Nama DNS publik instans atau alamat IP publik, yang bisa Anda dapatkan dari halaman detail instans.

- Nama pengguna —`Administrator`.

6. Saat klien meminta kredensil Anda, berikan kata sandi yang didekripsi dari Langkah 4.

Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Aplikasi AWS OpsWorks Stacks mewakili kode yang ingin Anda jalankan di server aplikasi. Kode itu sendiri berada di repositori seperti arsip Amazon S3; aplikasi berisi informasi yang diperlukan untuk menyebarkan kode ke instance server aplikasi yang sesuai.

Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks memicu peristiwa Deploy, yang menjalankan resep Deploy setiap lapisan. AWS OpsWorks Stacks juga menginstal konfigurasi stack dan atribut deployment yang berisi semua informasi yang diperlukan untuk menerapkan aplikasi, seperti repositori aplikasi dan data koneksi database.

Anda harus menerapkan resep khusus yang mengambil data penerapan aplikasi dari konfigurasi tumpukan dan atribut penerapan serta menangani tugas penerapan.

Topik

- Menambahkan Aplikasi
- Menerapkan Aplikasi
- Mengedit Aplikasi
- Menghubungkan Aplikasi ke Server Database
- Menggunakan Variabel Lingkungan
- Melewati Data ke Aplikasi
- Menggunakan Kunci SSH Repositori Git
- Menggunakan Domain Kustom

- Menggunakan SSL

Menambahkan Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Langkah pertama dalam menerapkan aplikasi ke server aplikasi Anda adalah menambahkan aplikasi ke tumpukan. Aplikasi mewakili aplikasi, dan berisi berbagai metadata, seperti nama dan jenis aplikasi, dan informasi yang diperlukan untuk menyebarkan aplikasi ke instance server, seperti URL repositori. Anda harus memiliki Kelola izin untuk menambahkan aplikasi ke tumpukan. Untuk informasi selengkapnya, lihat Mengelola izin.

> **Note**
>
> Prosedur di bagian ini berlaku untuk Chef 12 dan tumpukan yang lebih baru. Untuk informasi tentang cara menambahkan aplikasi ke layer di tumpukan Chef 11, lihatLangkah 2.4: Membuat dan Menyebarkan Aplikasi - Chef 11.

Untuk menambahkan aplikasi ke tumpukan

1. Letakkan kode di repositori pilihan Anda—arsip Amazon S3, repositori Git, repositori Subversion, atau arsip HTTP. Untuk informasi selengkapnya, lihat Sumber Aplikasi.
2. Klik Aplikasi di panel navigasi. Di halaman Aplikasi, klik Tambahkan aplikasi untuk aplikasi pertama Anda. Untuk aplikasi berikutnya, klik +Aplikasi.
3. Gunakan halaman App New untuk mengonfigurasi aplikasi, seperti yang dijelaskan di bagian berikut.

Mengonfigurasi Aplikasi

Halaman Tambah Aplikasi terdiri dari bagian berikut: Pengaturan, Sumber aplikasi, Sumber Data, Variabel Lingkungan, Tambah Domain, dan Pengaturan SSL.

Topik

- Pengaturan
- Sumber Aplikasi
- Sumber Data
- Variabel lingkungan
- Pengaturan Domain dan SSL

Pengaturan

Nama

Nama aplikasi, yang digunakan untuk mewakili aplikasi di UI. AWS OpsWorks Stacks juga menggunakan nama ini untuk menghasilkan nama pendek untuk aplikasi yang digunakan secara internal dan untuk mengidentifikasi aplikasi dalam konfigurasi tumpukan dan atribut penerapan. Setelah menambahkan aplikasi ke tumpukan, Anda dapat melihat nama pendeknya dengan mengklik Aplikasi di panel navigasi dan kemudian mengklik nama aplikasi untuk membuka halaman detail.

Akar dokumen

AWS OpsWorks Stacks menetapkan setelan root Dokumen ke `[:document_root]`atribut dalam atribut aplikasi. `deploy` Nilai default-nya adalah `null`. Resep penerapan Anda dapat memperoleh nilai tersebut dari `deploy` atribut menggunakan sintaks simpul Chef standar dan menyebarkan kode yang ditentukan ke lokasi yang sesuai di server. Untuk informasi selengkapnya tentang cara menerapkan aplikasi, lihatMenyebarkan Resep.

Sumber Aplikasi

Anda dapat menerapkan aplikasi dari jenis repositori berikut: Git, Amazon S3 bundle, HTTP bundle, dan Other. Semua jenis repositori mengharuskan Anda untuk menentukan jenis repositori dan URL repositori. Tipe repositori individu memiliki persyaratannya sendiri, seperti yang dijelaskan di bawah ini.

> **Note**
>
> AWS OpsWorks Stacks secara otomatis menyebarkan aplikasi dari repositori standar ke lapisan server bawaan. Jika Anda menggunakan jenis repositori lain, yang merupakan satu-satunya pilihan untuk tumpukan Windows, AWS OpsWorks Stacks menempatkan informasi repositori di `deploy`atribut aplikasi, tetapi Anda harus menerapkan resep khusus untuk menangani tugas penerapan.

Topik

- Arsip HTTP
- Arsip Amazon S3
- Repositori Git
- Repositori Lainnya

Arsip HTTP

Untuk menggunakan server HTTP yang dapat diakses publik sebagai repositori:

1. Buat arsip terkompresi—zip, gzip, bzip2, Java WAR, atau tarball—dari folder yang berisi kode aplikasi dan file terkait apa pun.

   > **Note**
   >
   > AWS OpsWorks Tumpukan tidak mendukung tarball yang tidak terkompresi.

2. Unggah file arsip ke server.
3. Untuk menentukan repositori di konsol, pilih HTTP Archive sebagai jenis repositori dan masukkan URL.

   Jika arsip dilindungi kata sandi, di bawah Sumber Aplikasi, tentukan kredenal masuk.

Arsip Amazon S3

Untuk menggunakan bucket Amazon Simple Storage Service sebagai repositori:

1. Buat bucket Amazon S3 publik atau pribadi. Untuk informasi selengkapnya, lihat Dokumentasi Amazon S3.
2. Agar AWS OpsWorks Stacks dapat mengakses bucket pribadi, Anda harus menjadi pengguna dengan setidaknya hak baca saja ke bucket Amazon S3 dan Anda akan memerlukan ID kunci akses dan kunci akses rahasia. Untuk informasi selengkapnya, lihat Dokumentasi AWS Identity and Access Management.
3. Masukkan kode dan file terkait dalam folder dan simpan folder dalam arsip terkompresi — zip, gzip, bzip2, Java WAR, atau tarball.

   > **Note**
   >
   > AWS OpsWorks Tumpukan tidak mendukung tarball yang tidak terkompresi.

4. Unggah file arsip ke bucket Amazon S3 dan rekam URL-nya.
5. Untuk menentukan repositori di konsol AWS OpsWorks Stacks, atur jenis Repositori ke Arsip S3 dan masukkan URL arsip. Untuk arsip pribadi, Anda juga harus memberikan ID kunci akses AWS dan kunci akses rahasia yang kebijakannya memberikan izin untuk mengakses bucket. Biarkan pengaturan ini kosong untuk arsip publik.

Repositori Git

Sebuah repositori Git menyediakan kontrol sumber dan pembuatan versi. AWS OpsWorks Stacks mendukung situs repositori yang dihosting secara publik seperti GitHubatau Bitbucket serta server Git yang dihosting secara pribadi. Untuk aplikasi dan submodul Git, format yang Anda gunakan untuk menentukan URL repositori di Sumber Aplikasi bergantung pada apakah repositori bersifat publik atau pribadi:

Repositori publik —Gunakan protokol hanya-baca HTTPS atau Git. Misalnya, Memulai dengan Chef 11 Linux Stacks menggunakan GitHub repositori publik yang dapat diakses oleh salah satu format URL berikut:

- Git hanya-baca: **`git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**
- HTTPS: **`https://github.com/amazonwebservices/opsworks-demo-php-simple-app.git`**

Repositori pribadi —Gunakan format baca/tulis SSH yang ditunjukkan dalam contoh ini:

- Repositori Github:. **`git@github.com:`*`project`*`/`*`repository`**
- Repositori di server Git: ***`user`*`@`*`server`*`:`*`project`*`/`*`repository`**

Memilih Git di bawah Kontrol Sumber menampilkan dua pengaturan opsional tambahan:

Kunci SSH repositori

Anda harus menentukan kunci SSH deploy untuk mengakses repositori Git pribadi. Bidang ini membutuhkan kunci pribadi; kunci publik ditetapkan ke repositori Git Anda. Untuk submodul Git, kunci yang ditentukan harus memiliki akses ke submodul tersebut. Untuk informasi selengkapnya, lihat Menggunakan Kunci SSH Repositori Git.

> **Important**
>
> Kunci SSH penerapan tidak dapat memerlukan kata sandi; AWS OpsWorks Tumpukan tidak memiliki cara untuk melewatinya.

Cabang/Revisi

Jika repositori memiliki beberapa cabang, AWS OpsWorks Stacks mengunduh cabang master secara default. Untuk menentukan cabang tertentu, masukkan nama cabang, SHA1 hash, atau nama tag. Untuk menentukan komit tertentu, masukkan pengenal komit 40 hekdigit penuh.

Repositori Lainnya

Jika repositori standar tidak memenuhi persyaratan Anda, Anda dapat menggunakan repositori lain, seperti Bazaar. Namun, AWS OpsWorks Stacks tidak secara otomatis menyebarkan aplikasi dari repositori tersebut. Anda harus menerapkan resep khusus untuk menangani proses penyebaran dan menetapkannya ke peristiwa Deploy lapisan yang sesuai. Untuk contoh cara menerapkan resep Deploy, lihatMenyebarkan Resep.

Sumber Data

Bagian ini melampirkan database ke aplikasi. Anda memiliki opsi berikut:

- RDS - Lampirkan salah satu lapisan layanan Amazon RDS stack.
- Tidak ada — Jangan melampirkan server database.

Jika Anda memilih RDS, Anda harus menentukan yang berikut ini.

Contoh basis data

Daftar ini mencakup setiap lapisan layanan Amazon RDS. Anda juga dapat memilih salah satu dari berikut ini:

(Wajib) Tentukan server database mana yang akan dilampirkan ke aplikasi. Isi daftar tergantung pada sumber data.

- RDS — Daftar lapisan layanan Amazon RDS stack.

Nama basis data

(Opsional) Tentukan nama database.

- Lapisan Amazon RDS — Masukkan nama database yang Anda tentukan untuk instans Amazon RDS.

  Anda bisa mendapatkan nama database dari konsol Amazon RDS.

Saat Anda menerapkan aplikasi dengan database terlampir, AWS OpsWorks Stacks menambahkan koneksi instance database ke atribut aplikasi. `deploy`

Anda dapat menulis resep khusus untuk mengambil informasi dari `deploy` atribut dan meletakkannya file yang dapat diakses oleh aplikasi. Ini adalah satu-satunya pilihan untuk menyediakan informasi koneksi database ke jenis aplikasi Lainnya.

Untuk informasi selengkapnya tentang cara menangani koneksi database, lihatMenghubungkan ke Database.

Untuk melepaskan server database dari aplikasi, edit konfigurasi aplikasi untuk menentukan server database yang berbeda, atau tidak ada server.

Variabel lingkungan

Anda dapat menentukan satu set variabel lingkungan untuk setiap aplikasi, yang khusus untuk aplikasi. Misalnya, jika Anda memiliki dua aplikasi, variabel lingkungan yang Anda tentukan untuk aplikasi pertama tidak tersedia untuk aplikasi kedua dan sebaliknya. Anda juga dapat menentukan variabel lingkungan yang sama untuk beberapa aplikasi dan menetapkan nilai yang berbeda untuk setiap aplikasi.

> **Note**
>
> Tidak ada batas tertentu pada jumlah variabel lingkungan. Namun, ukuran struktur data terkait—yang mencakup nama, nilai, dan nilai flag yang dilindungi variabel—tidak boleh melebihi 20 KB. Batas ini harus mengakomodasi sebagian besar jika tidak untuk semua kasus penggunaan. Melebihi itu akan menyebabkan kesalahan layanan (konsol) atau pengecualian (API) dengan pesan, “Lingkungan: terlalu besar (maksimum adalah 20 KB).”

AWS OpsWorks Stacks menyimpan variabel sebagai atribut dalam `deploy`atribut aplikasi. Anda dapat meminta resep kustom Anda mengambil nilai-nilai tersebut dengan menggunakan sintaks simpul Chef standar. Untuk contoh cara mengakses variabel lingkungan aplikasi, lihatMenggunakan Variabel Lingkungan.

Kunci

Nama variabel. Ini dapat berisi hingga 64 huruf besar dan kecil, angka, dan garis bawah (_), tetapi harus dimulai dengan huruf atau garis bawah.

Nilai

Nilai variabel. Ini dapat berisi hingga 256 karakter, yang semuanya harus dapat dicetak.

Nilai yang dilindungi

Apakah nilainya dilindungi. Pengaturan ini memungkinkan Anda menyembunyikan informasi sensitif seperti kata sandi. Jika Anda menetapkan nilai Protected untuk variabel, setelah Anda membuat aplikasi:

- Halaman detail aplikasi hanya menampilkan nama variabel, bukan nilainya.
- Jika Anda memiliki izin untuk mengedit aplikasi, Anda dapat mengklik Perbarui nilai untuk menentukan nilai baru, tetapi Anda tidak dapat melihat atau mengedit nilai lama.

> **Note**
>
> Log penerapan koki terkadang dapat menyertakan variabel lingkungan. Ini berarti variabel yang dilindungi mungkin ditampilkan di konsol. Untuk mencegah variabel yang dilindungi ditampilkan di konsol, kami sarankan Anda menggunakan bucket Amazon S3 sebagai penyimpanan untuk variabel yang dilindungi yang tidak ingin ditampilkan di konsol. Contoh

cara menggunakan bucket S3 untuk tujuan ini tersedia Menggunakan Amazon S3 Bucket dalam panduan ini.

Pengaturan Domain dan SSL

Untuk jenis aplikasi Lainnya, AWS OpsWorks Stacks menambahkan pengaturan ke `deploy` atribut aplikasi. Resep Anda dapat mengambil data dari atribut tersebut dan mengkonfigurasi server sesuai kebutuhan.

Pengaturan Domain

Bagian ini memiliki bidang Tambahkan Domain opsional untuk menentukan domain. Untuk informasi selengkapnya, lihat Menggunakan Domain Kustom.

Pengaturan SSL

Bagian ini memiliki sakelar Dukungan SSL yang dapat Anda gunakan untuk mengaktifkan atau menonaktifkan SSL. Jika Anda mengklik Ya, Anda harus memberikan informasi sertifikat SSL. Untuk informasi selengkapnya, lihat Menggunakan SSL.

Menerapkan Aplikasi

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

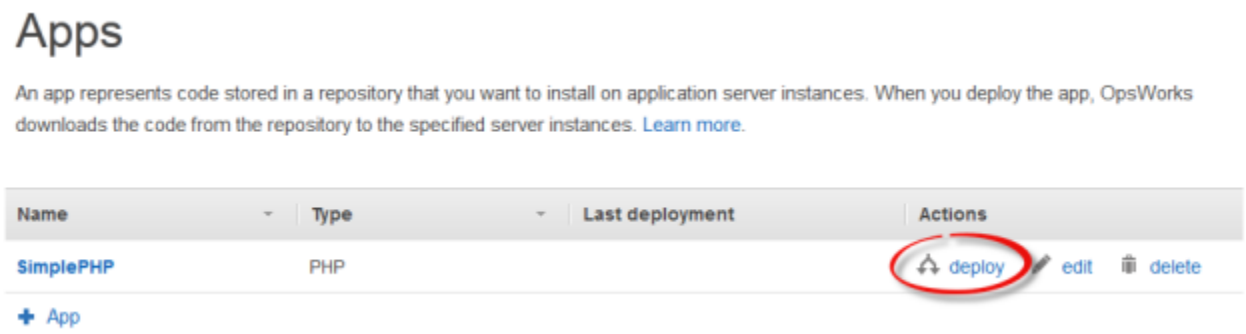
Tujuan utama penyebaran adalah untuk menyebarkan kode aplikasi dan file terkait ke instance server aplikasi. Operasi penyebaran ditangani oleh resep Deploy masing-masing instance, yang ditentukan oleh lapisan instance.

Saat Anda memulai sebuah instance, setelah resep Setup selesai, AWS OpsWorks Stacks secara otomatis menjalankan resep Deploy instance. Namun, saat menambahkan atau memodifikasi aplikasi, Anda harus menerapkannya secara manual ke instans online apa pun. Anda harus memiliki

izin Kelola atau Terapkan untuk menerapkan aplikasi. Untuk informasi selengkapnya, lihat Mengelola izin.

Untuk menerapkan aplikasi

1. Di halaman Aplikasi, klik tindakan penerapan aplikasi.



> **Note**
>
> Anda juga dapat menerapkan aplikasi dengan mengklik Deployment di panel navigasi. Pada halaman Deployment & Commands, klik Deploy an app Ketika Anda melakukan ini, Anda juga dapat memilih aplikasi mana yang akan digunakan.

2. Tentukan hal berikut:

   - (Wajib) Set Command: untuk menyebarkan, jika belum dipilih.
   - (Opsional) Sertakan komentar.

3. Klik Lanjutan >> untuk menentukan JSON kustom. AWS OpsWorks Stacks menambahkan satu set konfigurasi stack dan atribut deployment ke objek node. `deploy`Atribut berisi rincian penyebaran dan dapat digunakan oleh resep Deploy untuk menangani instalasi dan konfigurasi. Di tumpukan Linux, Anda dapat menggunakan bidang JSON khusus untuk mengganti pengaturan AWS OpsWorks Stacks default atau meneruskan pengaturan khusus ke resep kustom Anda. Untuk informasi selengkapnya tentang cara menggunakan JSON kustom, lihatMenggunakan JSON Kustom.

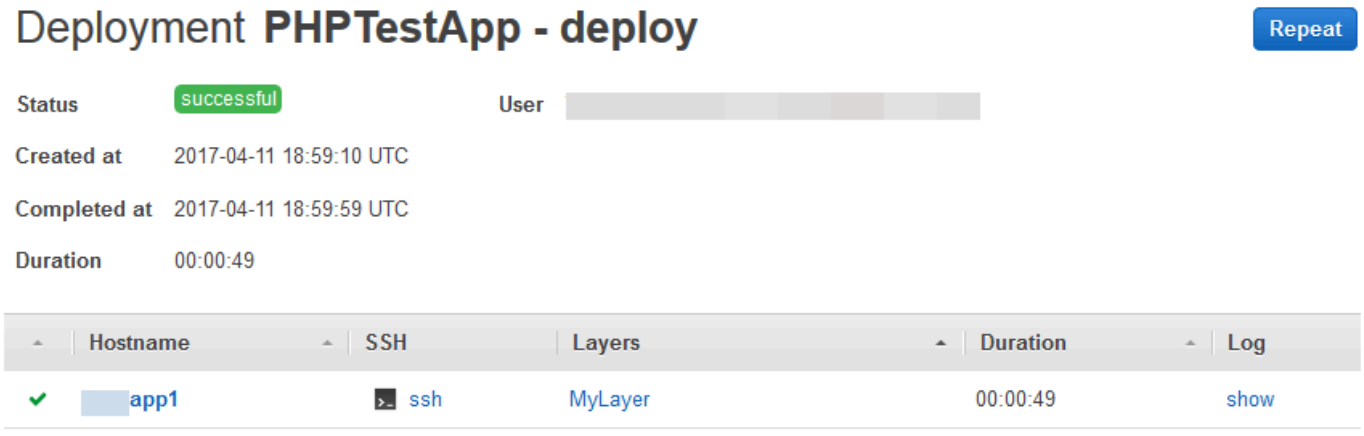
> **Note**
>
> Jika Anda menentukan JSON kustom di sini, itu ditambahkan ke konfigurasi tumpukan dan atribut penerapan hanya untuk penerapan ini. Jika Anda ingin menambahkan JSON kustom secara permanen, Anda harus menambahkannya ke tumpukan. Custom JSON

dibatasi hingga 120 KB. Jika Anda membutuhkan lebih banyak kapasitas, kami sarankan untuk menyimpan beberapa data di Amazon S3. Resep kustom Anda kemudian dapat menggunakan AWS CLI atau AWS SDK for Ruby untuk mengunduh data dari bucket ke instans Anda. Sebagai contoh, lihat Menggunakan SDK untuk Ruby.

4. Di bawah Instance, klik Advanced >> dan tentukan instance mana yang akan menjalankan perintah deploy.

   Perintah deploy memicu peristiwa Deploy, yang menjalankan resep penerapan pada instance yang dipilih. Resep penyebaran untuk server aplikasi terkait mengunduh kode dan file terkait dari repositori dan menginstalnya pada instance, sehingga Anda biasanya memilih semua instance server aplikasi terkait. Namun, jenis instance lain mungkin memerlukan beberapa perubahan konfigurasi untuk mengakomodasi aplikasi baru, jadi seringkali berguna untuk menjalankan resep penerapan pada instance tersebut juga. Resep-resep tersebut memperbarui konfigurasi sesuai kebutuhan tetapi tidak menginstal file aplikasi. Untuk informasi lebih lanjut tentang resep, lihat Buku Masak dan Resep.

5. Klik Deploy untuk menjalankan resep deploy pada instance yang ditentukan, yang menampilkan halaman Deployment. Saat proses selesai, AWS OpsWorks Stacks menandai aplikasi dengan tanda centang hijau untuk menunjukkan penerapan yang berhasil. Jika penerapan gagal, AWS OpsWorks Stacks menandai aplikasi dengan X merah. Dalam hal ini, Anda dapat membuka halaman Deployment dan memeriksa log penerapan untuk informasi selengkapnya.



Note

Saat Anda menerapkan pembaruan ke aplikasi JSP, Tomcat mungkin tidak mengenali pembaruan dan sebagai gantinya terus menjalankan versi aplikasi yang ada. Hal ini dapat terjadi, misalnya, jika Anda menerapkan aplikasi Anda sebagai file.zip yang hanya

berisi halaman JSP. Untuk memastikan bahwa Tomcat menjalankan versi yang paling baru digunakan, direktori root proyek harus menyertakan direktori WEB-INF yang berisi file. `web.xml` `web.xml`File dapat berisi berbagai konten, tetapi berikut ini cukup untuk memastikan bahwa Tomcat mengenali pembaruan dan menjalankan versi aplikasi yang saat ini digunakan. Anda tidak perlu mengubah versi untuk setiap pembaruan. Tomcat akan mengenali pembaruan meskipun versinya tidak berubah.

```
<context-param>
  <param-name>appVersion</param-name>
  <param-value>0.1</param-value>
</context-param>
```

Perintah Deployment Lainnya

Halaman aplikasi Deploy mencakup beberapa perintah lain untuk mengelola aplikasi Anda dan server terkait. Dari perintah berikut, hanya Undeploy yang tersedia untuk aplikasi di tumpukan Chef 12.

Undeploy

Memicu peristiwa Undeploy lifecycle, yang menjalankan resep undeploy untuk menghapus semua versi aplikasi dari instance yang ditentukan.

Rollback

Memulihkan versi aplikasi yang digunakan sebelumnya. Misalnya, jika Anda telah menerapkan aplikasi tiga kali dan kemudian menjalankan Rollback, server akan melayani aplikasi dari penerapan kedua. Jika Anda menjalankan Rollback lagi, server akan melayani aplikasi dari penerapan pertama. Secara default, AWS OpsWorks Stacks menyimpan lima penerapan terbaru, yang memungkinkan Anda untuk memutar kembali hingga empat versi. Jika Anda melebihi jumlah versi yang disimpan, perintah gagal dan meninggalkan versi tertua di tempatnya. Perintah ini tidak tersedia di tumpukan Chef 12.

Mulai Server Web

Menjalankan resep yang memulai server aplikasi pada instance yang ditentukan. Perintah ini tidak tersedia di tumpukan Chef 12.

Hentikan Server Web

Menjalankan resep yang menghentikan server aplikasi pada instance yang ditentukan. Perintah ini tidak tersedia di tumpukan Chef 12.

Mulai ulang Server Web

Menjalankan resep yang me-restart server aplikasi pada instance yang ditentukan. Perintah ini tidak tersedia di tumpukan Chef 12.

Important

Start Web Server, Stop Web Server, Restart Web Server, dan Rollback pada dasarnya adalah versi yang disesuaikan dari perintah stack Execute Recipes. Mereka menjalankan satu set resep yang melakukan tugas pada instance yang ditentukan.

- Perintah ini tidak memicu peristiwa siklus hidup, jadi Anda tidak dapat mengaitkannya untuk menjalankan kode khusus.
- Perintah ini hanya berfungsi untuk lapisan server aplikasi bawaan.

  Secara khusus, perintah ini tidak berpengaruh pada lapisan khusus, bahkan jika mereka mendukung server aplikasi. Untuk memulai, menghentikan, atau memulai ulang server pada lapisan khusus, Anda harus menerapkan resep khusus untuk melakukan tugas-tugas ini dan menggunakan perintah tumpukan Execute Recipes untuk menjalankannya. Untuk informasi selengkapnya tentang cara menerapkan dan menginstal resep khusus, lihatBuku Masak dan Resep.

Mengedit Aplikasi

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat mengubah konfigurasi aplikasi dengan mengedit aplikasi. Misalnya, jika Anda siap untuk menerapkan versi baru, Anda dapat mengedit pengaturan AWS OpsWorks Stacks aplikasi untuk menggunakan cabang repositori baru. Anda harus memiliki izin Kelola atau Terapkan untuk mengedit konfigurasi aplikasi. Untuk informasi selengkapnya, lihat Mengelola izin.

Untuk mengedit aplikasi

1. Pada halaman Aplikasi, klik nama aplikasi untuk membuka halaman detailnya.
2. Klik Edit untuk mengubah konfigurasi aplikasi.

   - Jika Anda mengubah nama aplikasi, AWS OpsWorks Stacks menggunakan nama baru untuk mengidentifikasi aplikasi di konsol.

     Mengubah nama tidak mengubah nama pendek yang terkait. Nama pendek disetel saat Anda menambahkan aplikasi ke tumpukan dan tidak dapat diubah selanjutnya.
   - Jika Anda telah menentukan variabel lingkungan yang dilindungi, Anda tidak dapat melihat atau mengedit nilainya. Namun, Anda dapat menentukan nilai baru dengan mengklik Perbarui nilai.
3. Klik Simpan untuk menyimpan konfigurasi baru dan kemudian Deploy App untuk menyebarkan aplikasi.

Mengedit aplikasi mengubah pengaturan dengan AWS OpsWorks Stacks, tetapi tidak memengaruhi instance tumpukan. Saat pertama kali menerapkan aplikasi, resep Deploy akan mengunduh kode dan file terkait ke instance server aplikasi, yang kemudian menjalankan salinan lokal. Jika Anda memodifikasi aplikasi di repositori atau mengubah setelan lainnya, Anda harus menerapkan aplikasi untuk menginstal pembaruan pada instance server aplikasi Anda, sebagai berikut. AWS OpsWorks Stacks secara otomatis menerapkan versi aplikasi saat ini ke instance baru saat dimulai. Namun, untuk kasus yang ada, situasinya berbeda:

- AWS OpsWorks Stacks secara otomatis menerapkan versi aplikasi saat ini ke instance baru saat dimulai.
- AWS OpsWorks Stacks secara otomatis menerapkan versi aplikasi terbaru ke instans offline, termasuk instans berbasis beban dan waktu, saat dimulai ulang.
- Anda harus menerapkan aplikasi yang diperbarui secara manual ke instans online.

Untuk informasi selengkapnya tentang cara menerapkan aplikasi, lihat Menerapkan Aplikasi

Menghubungkan Aplikasi ke Server Database

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat mengaitkan server database Amazon RDS dengan aplikasi saat membuat aplikasi atau yang lebih baru dengan mengedit aplikasi. Aplikasi Anda kemudian dapat menggunakan informasi koneksi database — nama pengguna, kata sandi,... —untuk terhubung ke server database. Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks memberikan informasi ini ke aplikasi dengan dua cara:

- Untuk tumpukan Linux, AWS OpsWorks Stacks membuat file pada setiap instance server aplikasi bawaan yang berisi data koneksi yang dapat digunakan aplikasi untuk terhubung ke server database.
- AWS OpsWorks Tumpukan menyertakan informasi koneksi dalam konfigurasi tumpukan dan atribut penerapan yang diinstal pada setiap instance.

  Anda dapat menerapkan resep khusus untuk mengekstrak informasi koneksi dari atribut ini dan memasukkannya ke dalam file dalam format pilihan Anda. Untuk informasi selengkapnya, lihat Melewati Data ke Aplikasi.

> **Important**
>
> Untuk tumpukan Linux, jika ingin mengaitkan lapisan layanan Amazon RDS dengan aplikasi, Anda harus menambahkan paket driver yang sesuai ke lapisan server aplikasi terkait, sebagai berikut:
>
> 1. Klik Layers di panel navigasi dan buka tab Resep server aplikasi.
> 2. Klik Edit dan tambahkan paket driver yang sesuai ke Paket OS. Misalnya, Anda harus menentukan `mysql` apakah layer berisi instance Amazon Linux dan `mysql-client` apakah layer tersebut berisi instance Ubuntu.

3. Simpan perubahan dan gunakan kembali aplikasi.

Menggunakan Resep Kustom

Anda dapat menerapkan resep kustom yang mengekstrak data koneksi dari `deploy`atribut aplikasi dan menyimpannya dalam bentuk yang dapat dibaca aplikasi, seperti file YAMG.

Anda melampirkan server database ke aplikasi saat membuat aplikasi atau yang lebih baru dengan mengedit aplikasi. Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks menginstal konfigurasi tumpukan dan atribut penerapan pada setiap instance yang menyertakan informasi koneksi database. Aplikasi Anda kemudian dapat mengambil atribut yang sesuai. Detailnya tergantung pada apakah Anda menggunakan tumpukan Linux atau Windows.

Menghubungkan ke Server Database untuk Linux Stack

Untuk tumpukan Linux, `deploy` namespace konfigurasi tumpukan dan atribut penerapan menyertakan atribut untuk setiap aplikasi yang di-deploy, dinamai dengan nama pendek aplikasi. Saat Anda melampirkan server database ke aplikasi, AWS OpsWorks Stacks mengisi `[:database]` atribut aplikasi dengan informasi koneksi, dan menginstalnya pada instance tumpukan untuk setiap penerapan berikutnya. Nilai atribut disediakan pengguna atau dihasilkan oleh AWS OpsWorks Stacks.

> **Note**
>
> AWS OpsWorks Stacks memungkinkan Anda untuk melampirkan server database ke beberapa aplikasi, tetapi setiap aplikasi hanya dapat memiliki satu server database terlampir. Jika Anda ingin menghubungkan aplikasi ke lebih dari satu server database, lampirkan salah satu server ke aplikasi, dan gunakan informasi dalam `deploy` atribut aplikasi untuk terhubung ke server tersebut. Gunakan JSON khusus untuk meneruskan informasi koneksi untuk server database lain ke aplikasi. Untuk informasi selengkapnya, lihat Melewati Data ke Aplikasi.

Aplikasi dapat menggunakan informasi koneksi dari `deploy` atribut instance untuk terhubung ke database. Namun, aplikasi tidak dapat mengakses informasi tersebut secara langsung—hanya resep yang dapat mengakses atribut. `deploy` Anda dapat mengatasi masalah ini dengan menerapkan resep khusus yang mengekstrak informasi koneksi dari `deploy` atribut dan memasukkannya ke dalam file yang dapat dibaca oleh aplikasi.

Menggunakan Variabel Lingkungan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Rekomendasi dalam topik ini berlaku untuk Chef 11.10 dan versi Chef sebelumnya. Untuk mendapatkan variabel lingkungan di Chef 12 dan rilis yang lebih baru, Anda harus menggunakan App Data Bag. Untuk informasi selengkapnya, lihat AWS OpsWorks Data Bag Referensi dan App Data Bag (aws_opsworks_app).

Saat Anda menentukan variabel lingkungan untuk aplikasi, AWS OpsWorks Stacks menambahkan definisi variabel ke `deploy`atribut aplikasi.

Lapisan kustom dapat menggunakan resep untuk mengambil nilai variabel dengan menggunakan sintaks node standar, dan menyimpannya dalam bentuk yang dapat diakses oleh aplikasi lapisan.

Anda harus menerapkan resep kustom yang memperoleh nilai variabel lingkungan dari `deploy` atribut instance. Resep kemudian dapat menyimpan data pada instance dalam bentuk yang dapat diakses oleh aplikasi, seperti file YAMG. Definisi variabel lingkungan aplikasi disimpan dalam `deploy` atribut, di aplikasi`environment_variables`. Contoh berikut menunjukkan lokasi atribut ini untuk aplikasi bernama`simplephpapp`, menggunakan JSON untuk mewakili struktur atribut.

```
{
  ...
  "ssh_users": {
  },
  "deploy": {
    "simplephpapp": {
      "application": "simplephpapp",
```

```
      "application_type": "php",
      "environment_variables": {
        "USER_ID": "168424",
        "USER_KEY": "somepassword"
      },
    ...
  }
}
```

Sebuah resep dapat memperoleh nilai variabel dengan menggunakan sintaks node standar. Contoh berikut menunjukkan cara mendapatkan `USER_ID` nilai dari JSON sebelumnya dan menempatkannya di log Chef.

```
Chef::Log.info("USER_ID: #{node[:deploy]['simplephpapp'][:environment_variables]
[:USER_ID]}")
```

Untuk penjelasan lebih rinci tentang cara mengambil informasi dari konfigurasi tumpukan dan penerapan JSON dan menyimpannya di instance, lihat. Melewati Data ke Aplikasi

Melewati Data ke Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Hal ini sering berguna untuk meneruskan data seperti pasangan kunci-nilai ke aplikasi di server. Untuk melakukannya, gunakan JSON khusus untuk menambahkan data ke tumpukan. AWS OpsWorks Stacks menambahkan data ke objek node setiap instance untuk setiap peristiwa siklus hidup.

Perhatikan, bagaimanapun, bahwa meskipun resep bisa mendapatkan data JSON kustom dari objek node dengan menggunakan atribut Chef, aplikasi tidak bisa. Salah satu pendekatan untuk mendapatkan data JSON kustom ke satu atau lebih aplikasi adalah dengan menerapkan resep khusus yang mengekstrak data dari node objek dan menulisnya ke file yang dapat dibaca

aplikasi. Contoh dalam topik ini menunjukkan cara menulis data ke file YAMG, tetapi Anda dapat menggunakan pendekatan dasar yang sama untuk format lain, seperti JSON atau XHTML.

Untuk meneruskan data nilai kunci ke instance tumpukan, tambahkan JSON khusus seperti berikut ini ke tumpukan. Untuk informasi selengkapnya tentang cara menambahkan JSON kustom ke tumpukan, lihatMenggunakan JSON Kustom.

```
{
  "my_app_data": {
    "app1": {
      "key1": "value1",
      "key2": "value2",
      "key3": "value3"
    },
    "app2": {
      "key1": "value1",
      "key2": "value2",
      "key3": "value3"
    }
  }
}
```

Contohnya mengasumsikan bahwa Anda memiliki dua aplikasi yang nama pendeknya `app1` dan`app2`, masing-masing memiliki tiga nilai data. Resep yang menyertainya mengasumsikan bahwa Anda menggunakan nama pendek aplikasi untuk mengidentifikasi data terkait; nama lainnya bersifat arbitrer. Untuk informasi selengkapnya tentang nama pendek aplikasi, lihatPengaturan.

Resep dalam contoh berikut menunjukkan cara mengekstrak data untuk setiap aplikasi dari `deploy` atribut dan memasukkannya ke dalam `.yml` file. Resepnya mengasumsikan bahwa JSON kustom Anda berisi data untuk setiap aplikasi.

```
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
```

`deploy`Atribut berisi atribut untuk setiap aplikasi, dinamai dengan nama pendek aplikasi. Setiap atribut aplikasi berisi sekumpulan atribut yang mewakili berbagai informasi tentang aplikasi. Contoh

ini menggunakan direktori penerapan aplikasi, yang diwakili oleh `[:deploy][:app_short_name][:deploy_to]` atribut. Untuk informasi selengkapnya tentang `[:deploy]`, lihat menyebarkan Atribut.

Untuk setiap aplikasi di`deploy`, resepnya melakukan hal berikut:

1. Membuat file bernama `app_data.yml` di `shared/config` subdirektori direktori aplikasi. `[:deploy_to]`

   Untuk informasi selengkapnya tentang cara AWS OpsWorks Stacks menginstal aplikasi, lihat. Menyebarkan Resep
2. Mengonversi nilai JSON kustom aplikasi ke YAMAL dan menulis data yang diformat ke. `app_data.yml`

Untuk meneruskan data ke aplikasi

1. Tambahkan aplikasi ke tumpukan dan catat nama pendeknya. Untuk informasi selengkapnya, lihat Menambahkan Aplikasi.
2. Tambahkan JSON khusus dengan data aplikasi ke `deploy` atribut, seperti yang dijelaskan sebelumnya. Untuk informasi selengkapnya tentang cara menambahkan JSON kustom ke tumpukan, lihatMenggunakan JSON Kustom.
3. Buat buku masak dan tambahkan resep ke dalamnya dengan kode berdasarkan contoh sebelumnya, dimodifikasi sesuai kebutuhan untuk nama atribut yang Anda gunakan di JSON kustom. Untuk informasi lebih lanjut tentang cara membuat buku masak dan resep, lihatBuku Masak dan Resep. Jika Anda sudah memiliki buku masak khusus untuk tumpukan ini, Anda juga dapat menambahkan resep ke buku masak yang ada, atau bahkan menambahkan kode ke resep Deploy yang ada.
4. Instal buku masak di tumpukan Anda. Untuk informasi selengkapnya, lihat Memasang Buku Masak Kustom.
5. Tetapkan resep ke peristiwa siklus hidup Deploy layer server aplikasi. AWS OpsWorks Tumpukan kemudian akan menjalankan resep pada setiap instance baru, setelah di-boot. Untuk informasi selengkapnya, lihat Resep Eksekusi.
6. Menerapkan aplikasi, yang juga menginstal konfigurasi tumpukan dan atribut penerapan yang sekarang berisi data Anda.

Note

Jika file data harus ada sebelum aplikasi di-deploy, Anda juga dapat menetapkan resep ke event siklus hidup Setup layer, yang terjadi sekali, tepat setelah instance selesai booting. Namun, AWS OpsWorks Stacks belum membuat direktori penerapan, jadi resep Anda harus membuat direktori yang diperlukan secara eksplisit sebelum membuat file data. Contoh berikut secara eksplisit membuat `/shared/config` direktori aplikasi, dan kemudian membuat file data di direktori tersebut.

```
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
```

Untuk memuat data, Anda dapat menggunakan sesuatu seperti kode Sinatra berikut:

```
#!/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
```

Anda dapat memperbarui nilai data aplikasi kapan saja dengan memperbarui JSON kustom, sebagai berikut.

Untuk memperbarui data aplikasi

1. Edit JSON kustom untuk memperbarui nilai data.
2. Terapkan aplikasi lagi, yang mengarahkan AWS OpsWorks Stacks untuk menjalankan resep Deploy pada instance tumpukan. Resep akan menggunakan atribut dari konfigurasi tumpukan yang diperbarui dan atribut penerapan, sehingga resep kustom Anda akan memperbarui file data dengan nilai saat ini.

Menggunakan Kunci SSH Repositori Git

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Kunci SSH repositori Git, kadang-kadang disebut kunci SSH deploy, adalah kunci SSH tanpa kata sandi yang menyediakan akses ke repositori Git pribadi. Idealnya, itu bukan milik pengembang tertentu. Tujuannya adalah untuk memungkinkan AWS OpsWorks Stacks untuk menyebarkan aplikasi atau buku masak secara asinkron dari repositori Git tanpa memerlukan masukan lebih lanjut dari Anda.

Berikut ini menjelaskan prosedur dasar untuk membuat kunci SSH repositori. Untuk detailnya, lihat dokumentasi untuk repositori Anda. Misalnya, Mengelola kunci penerapan menjelaskan cara membuat kunci SSH repositori untuk repositori, dan Kunci Deployment di Bitbucket menjelaskan cara membuat kunci SSH GitHub repositori untuk repositori Bitbucket. Perhatikan bahwa beberapa dokumentasi menjelaskan pembuatan kunci di server. Untuk AWS OpsWorks Stacks, cukup ganti “server” dengan “workstation” dalam instruksi.

Untuk membuat kunci SSH repositori

1. Buat deploy key pair SSH untuk repositori Git Anda di workstation Anda menggunakan program seperti. `ssh-keygen`

> **Important**
>
> AWS OpsWorks Stacks tidak mendukung frasa sandi kunci SSH.

2. Tetapkan kunci publik ke repositori dan simpan kunci pribadi di workstation Anda.
3. Masukkan kunci pribadi di kotak Kunci SSH Repositori saat Anda menambahkan aplikasi atau menentukan repositori buku masak. Untuk informasi selengkapnya, lihat Menambahkan Aplikasi.

AWS OpsWorks Tumpukan meneruskan kunci SSH repositori ke setiap instance, dan resep bawaan kemudian menggunakan kunci untuk terhubung ke repositori dan mengunduh kode. Kunci disimpan dalam `deploy`atribut sebagai `node[:deploy]['appshortname'][:scm][:ssh_key]`, dan hanya dapat diakses oleh pengguna root.

Menggunakan Domain Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika Anda meng-host nama domain dengan pihak ketiga, Anda dapat memetakan nama domain tersebut ke aplikasi. Prosedur dasarnya adalah sebagai berikut:

1. Buat subdomain dengan registrar DNS Anda dan petakan ke alamat IP Elastis penyeimbang beban atau alamat IP publik server aplikasi Anda.
2. Perbarui konfigurasi aplikasi Anda untuk menunjuk ke subdomain dan menerapkan ulang aplikasi.

> **Note**
>
> Pastikan Anda meneruskan nama domain yang tidak memenuhi syarat (seperti myapp1.example.com) ke nama domain yang memenuhi syarat (seperti www.myapp1.example.com) sehingga keduanya dipetakan ke aplikasi Anda.

Saat Anda mengonfigurasi domain untuk aplikasi, domain tersebut terdaftar sebagai alias server di file konfigurasi server. Jika Anda menggunakan penyeimbang beban, penyeimbang beban memeriksa nama domain di URL saat permintaan masuk dan mengarahkan lalu lintas berdasarkan domain.

Untuk memetakan subdomain ke alamat IP

1. Jika Anda menggunakan penyeimbang beban, pada halaman Instances, klik instans load balancer untuk membuka halaman detailnya dan mendapatkan alamat IP Elastis instans. Jika tidak, dapatkan alamat IP publik dari halaman detail instance server aplikasi.
2. Ikuti petunjuk yang diberikan oleh registrar DNS Anda untuk membuat dan memetakan subdomain Anda ke alamat IP dari Langkah 1.

> **Note**
>
> Jika instans penyeimbang beban berakhir di beberapa titik, Anda akan diberi alamat IP Elastis baru. Anda perlu memperbarui pengaturan registrar DNS Anda untuk memetakan ke alamat IP Elastis yang baru.

AWS OpsWorks Stacks hanya menambahkan pengaturan domain ke `deploy`atribut aplikasi. Anda harus menerapkan resep khusus untuk mengambil informasi dari objek node dan mengkonfigurasi server dengan tepat. Untuk informasi selengkapnya, lihat Buku Masak dan Resep.

Menjalankan Beberapa Aplikasi di Server Aplikasi yang Sama

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Informasi dalam topik ini tidak berlaku untuk aplikasi Node.js.

Jika Anda memiliki beberapa aplikasi dari jenis yang sama, terkadang lebih hemat biaya untuk menjalankannya pada instance server aplikasi yang sama.

Untuk menjalankan beberapa aplikasi di server yang sama

1. Tambahkan aplikasi ke tumpukan untuk setiap aplikasi.
2. Dapatkan subdomain terpisah untuk setiap aplikasi dan petakan subdomain ke server aplikasi atau alamat IP penyeimbang beban.
3. Edit konfigurasi setiap aplikasi untuk menentukan subdomain yang sesuai.

Untuk informasi selengkapnya tentang cara melakukan tugas-tugas ini, lihatMenggunakan Domain Kustom.

> **Note**
>
> Jika server aplikasi Anda menjalankan beberapa aplikasi HTTP, Anda dapat menggunakan Elastic Load Balancing untuk load-balancing. Untuk beberapa aplikasi HTTPS, Anda harus menghentikan koneksi SSL di penyeimbang beban atau membuat tumpukan terpisah untuk setiap aplikasi. Permintaan HTTPS dienkripsi, yang berarti bahwa jika Anda menghentikan koneksi SSL di server, penyeimbang beban tidak dapat memeriksa nama domain untuk menentukan aplikasi mana yang harus menangani permintaan tersebut.

Menggunakan SSL

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk menggunakan SSL dengan aplikasi Anda, Anda harus terlebih dahulu mendapatkan sertifikat server digital dari Certificate Authority (CA). Untuk kesederhanaan, panduan ini membuat sertifikat dan kemudian menandatanganinya sendiri. Sertifikat yang ditandatangani sendiri berguna untuk tujuan pembelajaran dan pengujian, tetapi Anda harus selalu menggunakan sertifikat yang ditandatangani oleh CA untuk tumpukan produksi.

Dalam panduan ini, Anda akan melakukan hal berikut:

1. Instal dan konfigurasi OpenSSL.
2. Buat kunci privat.
3. Buat permintaan penandatanganan sertifikat.
4. Hasilkan sertifikat yang ditandatangani sendiri.
5. Edit aplikasi dengan informasi sertifikat Anda.

> **Important**
>
> Jika aplikasi Anda menggunakan SSL, kami sarankan Anda menonaktifkan, jika mungkin SSLv3, di lapisan server aplikasi Anda untuk mengatasi kerentanan yang dijelaskan dalam CVE-2014-3566. Jika tumpukan Anda menyertakan lapisan Ganglia, Anda harus menonaktifkan SSL v3 untuk lapisan itu juga. Detailnya tergantung pada lapisan tertentu; untuk informasi lebih lanjut, lihat yang berikut ini.
>
> - Lapisan AWS OpsWorks Tumpukan Server Aplikasi Java
> - Lapisan AWS OpsWorks Tumpukan Server Aplikasi Node.js
> - Lapisan AWS OpsWorks Tumpukan Server Aplikasi PHP

- Lapisan AWS OpsWorks Tumpukan Server Aplikasi Rails
- Lapisan AWS OpsWorks Tumpukan Server Web Statis
- Lapisan Ganglia

Topik

- Langkah 1: Instal dan Konfigurasikan OpenSSL
- Langkah 2: Buat Kunci Pribadi
- Langkah 3: Buat Permintaan Penandatanganan Sertifikat
- Langkah 4: Kirim CSR ke Otoritas Sertifikat
- Langkah 5: Edit Aplikasi

Langkah 1: Instal dan Konfigurasikan OpenSSL

Membuat dan mengunggah sertifikat server memerlukan alat yang mendukung protokol SSL dan TLS. OpenSSL adalah alat sumber terbuka yang menyediakan fungsi kriptografi dasar yang diperlukan untuk membuat token RSA dan menandatanganinya dengan kunci pribadi Anda.

Prosedur berikut mengasumsikan bahwa komputer Anda belum menginstal OpenSSL.

Untuk menginstal OpenSSL di Linux dan Unix

1. Pergi ke OpenSSL: Sumber, Tarballs.
2. Unduh sumber terbaru.
3. Bangun paketnya.

Untuk menginstal OpenSSL di Windows

1. Jika Microsoft Visual C ++ 2008 Redistributable Package belum diinstal pada sistem Anda, download paket.
2. Jalankan installer dan ikuti petunjuk yang diberikan oleh Microsoft Visual C ++ 2008 Redistributable Setup Wizard untuk menginstal redistributable.
3. Buka OpenSSL: Distribusi Biner, klik versi binari OpenSSL yang sesuai untuk lingkungan Anda, dan simpan penginstal secara lokal.
4. Jalankan installer dan ikuti instruksi di OpenSSL Setup Wizard untuk menginstal binari.

Buat variabel lingkungan yang menunjuk ke titik instalasi OpenSSL dengan membuka terminal atau jendela perintah dan menggunakan baris perintah berikut.

- Di Linux dan Unix

```
export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

- Pada Windows

```
set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

Tambahkan path binari OpenSSL ke variabel path komputer Anda dengan membuka terminal atau jendela perintah dan menggunakan baris perintah berikut.

- Di Linux dan Unix

```
export PATH=$PATH:$OpenSSL_HOME/bin
```

- Pada Windows

```
set Path=OpenSSL_HOME\bin;%Path%
```

> **Note**
>
> Setiap perubahan yang Anda buat pada variabel lingkungan dengan menggunakan baris perintah ini hanya valid untuk sesi baris perintah saat ini.

Langkah 2: Buat Kunci Pribadi

Anda memerlukan kunci pribadi unik untuk membuat Permintaan Penandatanganan Sertifikat (CSR). Buat kunci dengan menggunakan baris perintah berikut:

```
openssl genrsa 2048 > privatekey.pem
```

Langkah 3: Buat Permintaan Penandatanganan Sertifikat

Permintaan Penandatanganan Sertifikat (CSR) adalah file yang dikirim ke Otoritas Sertifikat (CA) untuk mengajukan sertifikat server digital. Buat CSR dengan menggunakan baris perintah berikut.

```
openssl req -new -key privatekey.pem -out csr.pem
```

Output perintah akan terlihat mirip dengan berikut ini:

```
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.
```

Tabel berikut dapat membantu Anda membuat permintaan sertifikat.

Data Permintaan Sertifikat

Nama	Deskripsi	Contoh
Nama Negara	Singkatan ISO dua huruf untuk negara Anda.	AS = Amerika Serikat
Negara Bagian atau Provinsi	Nama negara bagian atau provinsi tempat organisasi Anda berada. Nama ini tidak boleh disingkat.	Washington
Nama Lokal	Nama kota tempat organisasi Anda berada.	Seattle
Nama Organisasi	Nama lengkap legal organisasi Anda. Jangan menyingkat nama organisasi Anda.	CorporationX
Unit Organisasi	(Opsional) Untuk informasi organisasi tambahan.	Pemasaran

Nama	Deskripsi	Contoh
Nama Umum	Nama domain yang sepenuhnya memenuhi syarat untuk CNAME Anda. Anda akan menerima peringatan pemeriksaan nama sertifikat jika ini tidak sama persis.	www.example.com
Alamat Email	Alamat email administrator server	someone@example.com

> **Note**
>
> Bidang Nama Umum sering disalahpahami dan diselesaikan secara tidak benar. Nama umum biasanya host Anda ditambah nama domain. Ini akan terlihat seperti "www.example.com" atau "example.com". Anda perlu membuat CSR menggunakan nama umum yang benar.

Langkah 4: Kirim CSR ke Otoritas Sertifikat

Untuk penggunaan produksi, Anda akan memperoleh sertifikat server dengan mengirimkan CSR Anda ke Otoritas Sertifikat (CA), yang mungkin memerlukan kredensi atau bukti identitas lainnya. Jika aplikasi Anda berhasil, CA mengembalikan sertifikat identitas yang ditandatangani secara digital dan mungkin file rantai sertifikat. AWS tidak merekomendasikan CA tertentu. Untuk sebagian daftar yang tersedia CAs, lihat Otoritas Sertifikat - Penyedia di Wikipedia.

Anda juga dapat membuat sertifikat yang ditandatangani sendiri, yang hanya dapat digunakan untuk tujuan pengujian. Untuk contoh ini, gunakan baris perintah berikut untuk menghasilkan sertifikat yang ditandatangani sendiri.

```
openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out server.crt
```

Output akan terlihat serupa dengan yang berikut ini:

```
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
```

Langkah 5: Edit Aplikasi

Setelah Anda membuat sertifikat dan menandatanganinya, perbarui aplikasi Anda untuk mengaktifkan SSL dan memberikan informasi sertifikat Anda. Di halaman Aplikasi, pilih aplikasi untuk membuka halaman detail, lalu klik Edit Aplikasi. Untuk mengaktifkan dukungan SSL, atur Aktifkan SSL ke Ya, yang menampilkan opsi konfigurasi berikut.

Sertifikat SSL

Tempelkan isi file public key certificate (.crt) ke dalam kotak. Sertifikat harus terlihat seperti berikut:

```
-----BEGIN CERTIFICATE-----
MIICuTCCAiICCQCtqFKItVQJpzANBgkqhkiG9w0BAQUFADCBoDELMAkGA1UEBhMC
dXMxEzARBgNVBAgMCndhc2hpbmd0b24xEDAOBgNVBAcMB3NlYXR0bGUxDzANBgNV
BAoMBmFtYXpvbjEWMBQGA1UECwwNRGV2IGFuZCBUb29sczEdMBsGA1UEAwwUc3Rl
cGhhbmllYXBpZXJjZS5jb20xIjAgBgkqhkiG9w0BCQEWE3NhcGllcmNlQGFtYXpv
...
-----END CERTIFICATE-----
```

> **Note**
>
> Jika Anda menggunakan Nginx dan Anda memiliki file rantai sertifikat, Anda harus menambahkan konten ke file sertifikat kunci publik.

Jika Anda memperbarui sertifikat yang ada, lakukan hal berikut:

- Pilih Perbarui sertifikat SSL untuk memperbarui sertifikat.
- Jika sertifikat baru tidak cocok dengan kunci pribadi yang ada, pilih Perbarui kunci sertifikat SSL.
- Jika sertifikat baru tidak cocok dengan rantai sertifikat yang ada, pilih Perbarui sertifikat SSL.

Kunci Sertifikat SSL

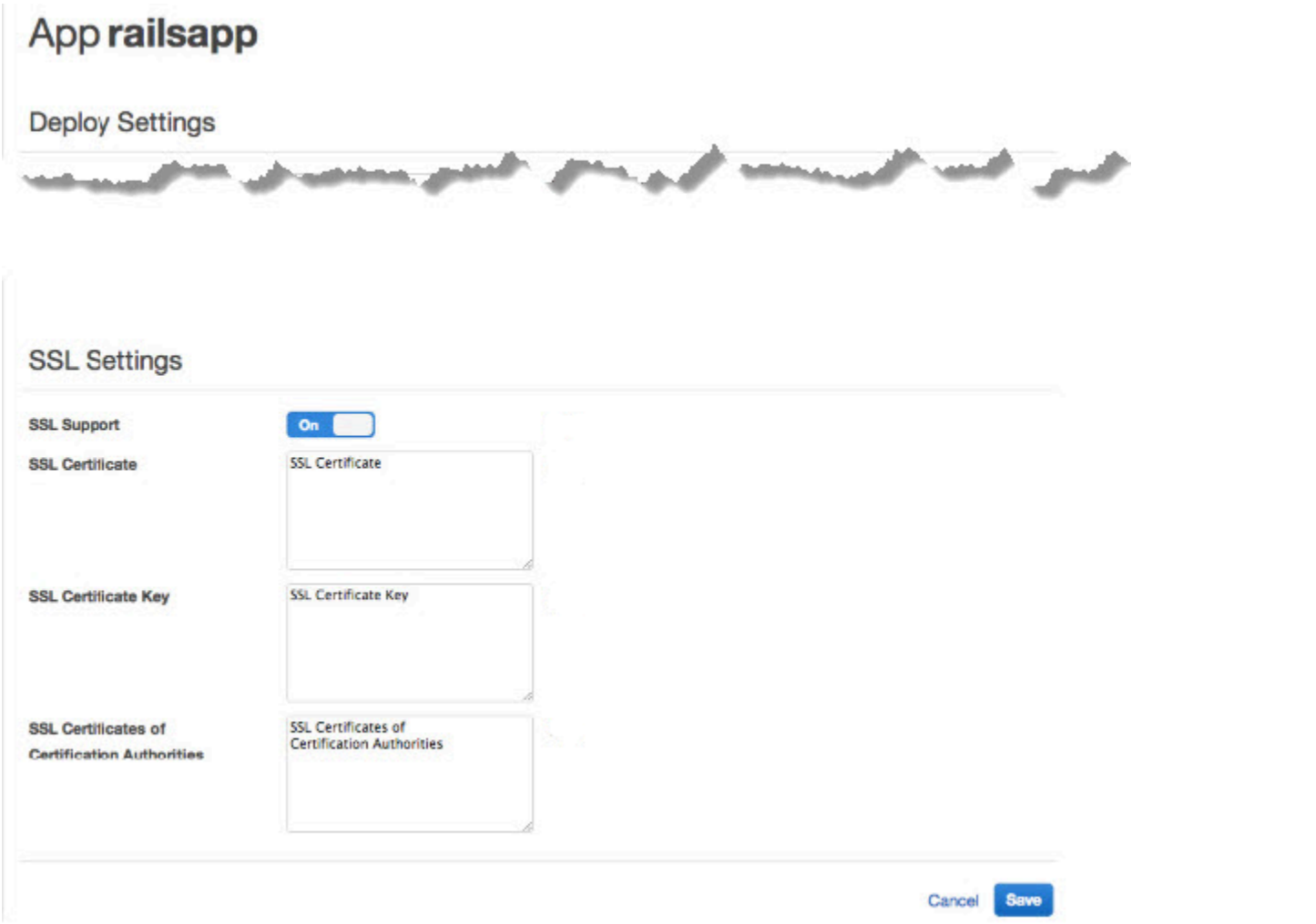
Tempelkan isi file kunci pribadi (file.pem) ke dalam kotak. Seharusnya terlihat seperti berikut:

```
----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQC0CYklJY5r4vV2NHQYEpwtsLuMMBhylMrgBShKq+HHVLYQQCL6
+wGIiRq5qXqZlRXje3GM5Jvcm6q0R71MfRIl1FuzKyqDtneZaAIEYniZibHiUnmO
/UNqpFDosw/6hY3ONk0fSBlU4ivD0Gjpf6J80jL3DJ4R23Ed0sdL4pRT3QIDAQAB
AoGBAKmMfWrNRqYVtGKgnWB6Tji9QrKQLMXjmHeGg95mppdJELiXHhpMvrHtpIyK
...
-----END RSA PRIVATE KEY-----
```

Sertifikat SSL dari Otoritas Sertifikasi

Jika Anda memiliki file rantai sertifikat, tempelkan konten ke dalam kotak.

> **Note**
>
> Jika Anda menggunakan Nginx, Anda harus membiarkan kotak ini kosong. Jika Anda memiliki file rantai sertifikat, tambahkan ke file sertifikat kunci publik di Sertifikat SSL.



Setelah Anda mengklik Simpan, gunakan ulang aplikasi untuk memperbarui instans online Anda.

Untuk lapisan server aplikasi bawaan, AWS OpsWorks Stacks secara otomatis memperbarui konfigurasi server. Setelah penerapan selesai, Anda dapat memverifikasi bahwa instalasi OpenSSL Anda berfungsi, sebagai berikut.

Untuk memverifikasi instalasi OpenSSL

1. Pergi ke halaman Instances.
2. Jalankan aplikasi dengan mengklik alamat IP instans server aplikasi atau, jika Anda menggunakan penyeimbang beban, alamat IP penyeimbang beban.
3. Ubah awalan alamat IP dari **`http://`** ke **`https://`** dan segarkan browser untuk memverifikasi halaman dimuat dengan benar dengan SSL.

Pengguna yang telah mengonfigurasi aplikasi untuk dijalankan di Mozilla Firefox terkadang mendapatkan kesalahan sertifikat berikut:`SEC_ERROR_UNKNOWN_ISSUER`. Kesalahan ini dapat disebabkan oleh fungsionalitas penggantian sertifikat dalam program antivirus dan antimalware organisasi Anda, oleh beberapa jenis perangkat lunak pemantauan dan pemfilteran lalu lintas jaringan, atau oleh malware. Untuk informasi selengkapnya tentang cara memecahkan masalah kesalahan ini, lihat Cara memecahkan masalah kode kesalahan keamanan di situs web aman di situs web Dukungan Mozilla Firefox.

Untuk semua lapisan lain, termasuk lapisan khusus, AWS OpsWorks Stacks hanya menambahkan pengaturan SSL ke atribut aplikasi. `deploy` Anda harus menerapkan resep khusus untuk mengambil informasi dari objek node dan mengkonfigurasi server dengan tepat.

Buku Masak dan Resep

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks menggunakan buku masak Chef untuk menangani tugas-tugas seperti menginstal dan mengonfigurasi paket dan menerapkan aplikasi. Bagian ini menjelaskan cara menggunakan buku masak dengan AWS OpsWorks Tumpukan. Untuk informasi selengkapnya, lihat Chef.

> **Note**
>
> AWS OpsWorks Stacks saat ini mendukung Chef versi 12, 11.10.4, 11.4.4, dan 0.9.15.5. Namun, Chef 0.9.15.5 sudah usang dan kami tidak menyarankan Anda menggunakannya untuk tumpukan baru. Untuk kenyamanan, mereka biasanya disebut hanya dengan nomor versi mayor dan minor mereka. Tumpukan yang menjalankan Chef 0.9 atau 11.4 menggunakan Chef Solo dan tumpukan yang menjalankan Chef 12 atau 11.10 menggunakan Chef Client dalam mode lokal. Untuk tumpukan Linux, Anda dapat menggunakan Manajer Konfigurasi untuk menentukan versi Chef mana yang akan digunakan saat Anda membuat tumpukan. Tumpukan Windows harus menggunakan Chef 12.2. Untuk informasi

selengkapnya, termasuk panduan untuk memigrasikan tumpukan ke versi Chef yang lebih baru, lihat. Versi Chef

Topik

- Repositori
- Versi Chef
- Versi Ruby
- Memasang Buku Masak Kustom
- Memperbarui Buku Masak Kustom
- Resep Eksekusi

Repositori

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Buku masak kustom Anda harus disimpan dalam repositori online, baik arsip seperti file.zip atau manajer kontrol sumber seperti Git. Tumpukan hanya dapat memiliki satu repositori buku masak khusus, tetapi repositori dapat berisi sejumlah buku masak. Saat Anda menginstal atau memperbarui buku masak, AWS OpsWorks Stacks menginstal seluruh repositori dalam cache lokal pada setiap instance tumpukan. Ketika sebuah instance membutuhkan, misalnya, untuk menjalankan satu atau lebih resep, ia menggunakan kode dari cache lokal.

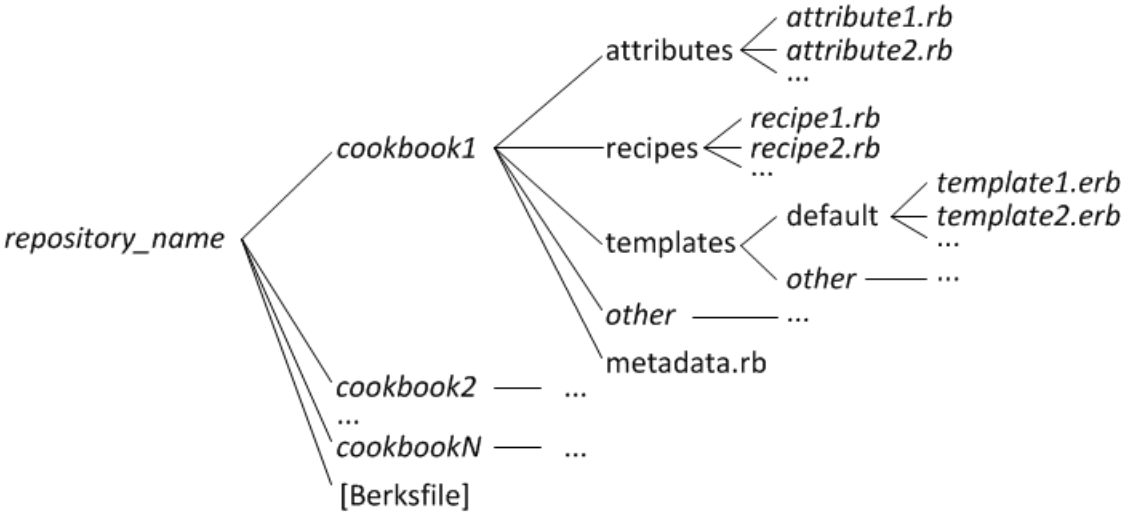
Berikut ini menjelaskan cara menyusun repositori buku masak Anda, yang tergantung pada jenisnya. Teks yang dicetak miring dalam ilustrasi mewakili direktori dan nama file yang ditentukan pengguna, termasuk nama repositori atau arsip.

Manajer Kontrol Sumber

AWS OpsWorks Stacks mendukung manajer kontrol sumber berikut:

- Tumpukan Linux — Git dan Subversion
- Tumpukan Windows - Git

Berikut ini menunjukkan direktori dan struktur file yang diperlukan:



- Direktori buku masak semuanya harus berada di tingkat atas.

Arsip

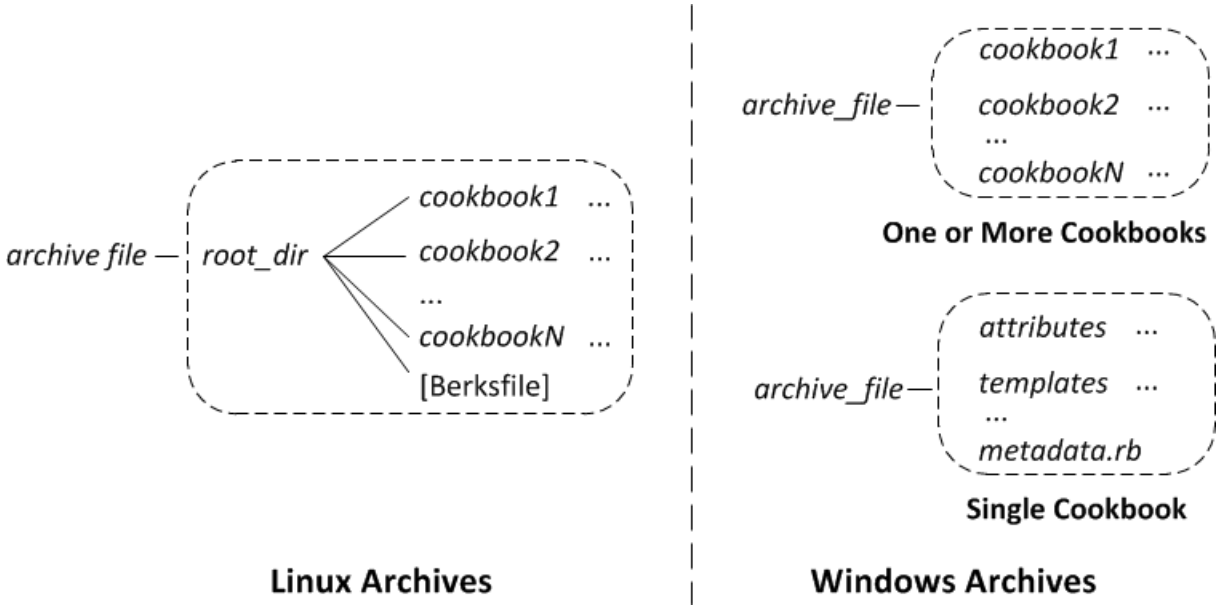
AWS OpsWorks Stacks mendukung arsip berikut:

- Tumpukan Linux — file zip, gzip, bzip2, atau tarball, disimpan di Amazon S3 atau situs web (arsip HTTP).

  AWS OpsWorks Tumpukan tidak mendukung tarball yang tidak terkompresi.

- Tumpukan Windows — file zip dan tgz (tar terkompresi gzip), disimpan di Amazon S3.

Berikut ini menunjukkan direktori dan struktur file yang diperlukan, yang tergantung pada apakah Anda menjalankan tumpukan Linux atau Windows. Struktur buku masak sama dengan repositori SCM, sehingga diwakili oleh elipsis (...).



- Tumpukan Linux — Direktori buku masak harus terkandung dalam direktori root.
- Tumpukan Windows — Buku masak harus berada di tingkat atas arsip.

  Jika Anda hanya memiliki satu buku masak, Anda dapat secara opsional menghilangkan direktori buku masak dan meletakkan file buku masak di tingkat atas. Dalam hal ini, AWS OpsWorks Stacks memperoleh nama buku masak dari metadata.rb.

Setiap direktori buku masak memiliki setidaknya satu dan biasanya semua direktori dan file standar berikut, yang harus menggunakan nama standar:

- `attributes`— File atribut buku masak.
- `recipes`— File resep buku masak.
- `templates`— File templat buku masak.
- *`other`*— Direktori yang ditentukan pengguna opsional yang berisi jenis file lain, seperti definisi atau spesifikasi.
- `metadata.rb`— Metadata buku masak.

  Untuk Chef 11.10 dan yang lebih baru, jika resep Anda bergantung pada buku masak lain, Anda harus menyertakan `depends` pernyataan yang sesuai dalam file buku masak Anda. `metadata.rb` Misalnya, jika buku masak Anda menyertakan resep dengan pernyataan seperti`include_recipe anothercookbook::somerecipe`, `metadata.rb` file buku masak Anda harus menyertakan baris berikut: `depends "anothercookbook"` Untuk informasi lebih lanjut, lihat Tentang Metadata Buku Masak.

Template harus dalam subdirektori `templates` direktori, yang berisi setidaknya satu dan opsional beberapa subdirektori. Subdirektori tersebut secara opsional dapat memiliki subdirektori juga.

- Template biasanya memiliki `default` subdirektori, yang berisi file template yang digunakan Chef secara default.
- lain mewakili subdirektori opsional yang dapat digunakan untuk template khusus sistem operasi.
- Chef secara otomatis menggunakan template dari subdirektori yang sesuai, berdasarkan konvensi penamaan yang dijelaskan dalam Kekhususan File. Misalnya, untuk sistem operasi Amazon Linux dan Ubuntu, Anda dapat menempatkan template khusus sistem operasi di subdirektori bernama `amazon` atau, masing-masing. `ubuntu`

Detail bagaimana Anda menangani buku masak khusus bergantung pada jenis repositori pilihan Anda.

Untuk menggunakan arsip

1. Terapkan buku masak Anda dengan menggunakan struktur folder yang ditunjukkan di bagian sebelumnya.
2. Buat arsip terkompresi dan unggah ke bucket Amazon S3 atau situs web.

   Jika Anda memperbarui buku masak Anda, Anda harus membuat dan mengunggah file arsip baru. Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang cara menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Cara Menghapus Bucket S3? .

Untuk menggunakan SCM

1. Siapkan repositori Git atau Subversion menggunakan struktur yang ditunjukkan sebelumnya.
2. Secara opsional, gunakan fitur kontrol versi repositori untuk mengimplementasikan beberapa cabang atau versi.

   Jika Anda memperbarui buku masak Anda, Anda dapat melakukannya di cabang baru dan langsung OpsWorks menggunakan versi baru. Anda juga dapat menentukan versi tagged tertentu. Lihat perinciannya di Mencantumkan Repositori Cookbook Kustom.

Memasang Buku Masak Kustommenjelaskan cara agar AWS OpsWorks Stacks menginstal repositori buku masak Anda pada instance tumpukan.

> **Important**
>
> Setelah Anda memperbarui buku masak yang ada di repositori, Anda harus menjalankan perintah `update_cookbooks` stack untuk mengarahkan AWS OpsWorks Stacks untuk memperbarui cache lokal setiap instans online. Untuk informasi selengkapnya, lihat Perintah Tumpukan.

Versi Chef

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks mendukung beberapa versi Chef. Anda memilih versi saat Anda membuat tumpukan. AWS OpsWorks Stacks kemudian menginstal versi Chef itu di semua instance tumpukan bersama dengan satu set resep bawaan yang kompatibel dengan versi itu. Jika Anda menginstal resep khusus apa pun, resep tersebut harus kompatibel dengan versi Chef tumpukan.

AWS OpsWorks Tumpukan saat ini mendukung Chef versi 12, 11.10, 11.4 dan 0.9 untuk tumpukan Linux, dan Chef 12.2 (saat ini Chef 12.22) untuk tumpukan Windows. Untuk kenyamanan, mereka biasanya disebut hanya dengan nomor versi mayor dan minor mereka. Untuk tumpukan Linux, Anda dapat menggunakan Manajer Konfigurasi untuk menentukan versi Chef mana yang akan digunakan saat Anda membuat tumpukan. Tumpukan Windows harus menggunakan Chef 12.2. Untuk informasi selengkapnya, termasuk panduan untuk memigrasikan tumpukan ke versi Chef yang lebih baru, lihat. Versi Chef Untuk informasi versi lengkap, lihatAWS OpsWorks Stacks sistem operasi.

Koki 12.2

Dukungan Chef 12.2 diperkenalkan pada Mei 2015, dan hanya digunakan oleh tumpukan Windows. Versi Chef saat ini di tumpukan Windows adalah Chef 12.22. Ini berjalan dengan Ruby 2.3.6, dan menggunakan chef-client dalam mode lokal, yang meluncurkan server Chef dalam memori lokal yang disebut chef-zero. Kehadiran server ini memungkinkan resep untuk

menggunakan pencarian Chef dan tas data. Dukungan memiliki beberapa keterbatasan, yang dijelaskan dalamResep Penerapan: Chef 12.2, tetapi Anda dapat menjalankan banyak buku masak komunitas tanpa modifikasi.

Koki 12

Dukungan Chef 12 diperkenalkan pada bulan Desember 2015, dan hanya digunakan oleh tumpukan Linux. Ini berjalan dengan Ruby 2.1.6 atau 2.2.3 dan menggunakan chef-client dalam mode lokal, yang memungkinkan resep untuk menggunakan pencarian Chef dan tas data. Untuk informasi selengkapnya, lihat AWS OpsWorks Stacks sistem operasi.

Koki 11.10

Dukungan Chef 11.10 diperkenalkan pada Maret 2014, dan hanya digunakan oleh tumpukan Linux. Ini berjalan dengan Ruby 2.0.0 dan menggunakan chef-client dalam mode lokal, yang memungkinkan resep untuk menggunakan pencarian Chef dan tas data. Dukungan memiliki beberapa keterbatasan, yang dijelaskan dalamResep Penerapan: Chef 11.10, tetapi Anda dapat menjalankan banyak buku masak komunitas tanpa modifikasi. Anda juga dapat menggunakan Berkshelf untuk mengelola dependensi buku masak Anda. Versi Berkshelf yang didukung tergantung pada sistem operasi. Untuk informasi selengkapnya, lihat AWS OpsWorks Stacks sistem operasi. Anda tidak dapat membuat tumpukan CentOS yang menggunakan Chef 11.10.

Koki 11.4

Dukungan Chef 11.4 diperkenalkan pada Juli 2013, dan hanya digunakan oleh tumpukan Linux. Ini berjalan dengan Ruby 1.8.7 dan menggunakan chef-solo, yang tidak mendukung pencarian Chef atau tas data. Anda sering dapat menggunakan buku masak komunitas yang bergantung pada fitur-fitur tersebut dengan AWS OpsWorks Stacks, tetapi Anda harus memodifikasinya seperti yang dijelaskan dalam. Bermigrasi ke Versi Chef baru Anda tidak dapat membuat tumpukan CentOS yang menggunakan Chef 11.4. Tumpukan Chef 11.4 tidak didukung di endpoint regional di luar Wilayah US East (N. Virginia).

Koki 0.9

Chef 0.9 hanya digunakan oleh tumpukan Linux dan tidak lagi didukung. Perhatikan detail ini:

- Anda tidak dapat menggunakan konsol untuk membuat tumpukan Chef 0.9 baru.

  Anda harus menggunakan CLI atau API, atau Anda harus membuat tumpukan dengan versi Chef yang berbeda dan kemudian mengedit konfigurasi tumpukan.

- Fitur AWS OpsWorks Stacks baru tidak tersedia untuk tumpukan Chef 0.9.

- Versi sistem operasi baru hanya akan memberikan dukungan terbatas untuk tumpukan Chef 0.9.

  Secara khusus, Amazon Linux 2014.09 dan versi yang lebih baru tidak mendukung tumpukan Chef 0.9 dengan lapisan Rails App Server yang bergantung pada Ruby 1.8.7.
- Wilayah AWS baru, termasuk Eropa (Frankfurt), tidak mendukung tumpukan Chef 0.9.

> **Note**
>
> Kami tidak merekomendasikan penggunaan Chef 0.9 untuk tumpukan baru. Anda harus memigrasikan tumpukan yang ada ke versi Chef terbaru sesegera mungkin.

Jika Anda ingin menggunakan buku masak komunitas dengan AWS OpsWorks Stacks, kami sarankan Anda menentukan Chef 12 untuk tumpukan Linux baru dan memigrasikan tumpukan Linux yang ada ke Chef 12. Anda dapat menggunakan konsol AWS OpsWorks Stacks, API, atau CLI untuk memigrasikan tumpukan yang ada ke versi Chef yang lebih baru. Untuk informasi selengkapnya, lihat Bermigrasi ke Versi Chef baru.

Topik

- Menerapkan Resep untuk Tumpukan Chef 12.2
- Menerapkan Resep untuk Chef 12 Stacks
- Menerapkan Resep untuk Tumpukan Chef 11.10
- Menerapkan Resep untuk Chef 11.4 Stacks
- Migrasi Stack Linux yang Ada ke Versi Chef baru

Menerapkan Resep untuk Tumpukan Chef 12.2

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Chef 12.2 (saat ini Chef 12.22) hanya tersedia di tumpukan Windows, yang harus menjalankan versi Chef tersebut.

- Resep harus menggunakan atribut dan sumber daya khusus Windows untuk beberapa tujuan.

  Untuk informasi lebih lanjut, lihat Chef untuk Microsoft Windows.
- Chef menjalankan menggunakan Ruby 2.3.6, sehingga resep Anda dapat menggunakan sintaks Ruby baru.
- Resep dapat menggunakan pencarian Chef dan tas data.

  Tumpukan Chef 12.2 dapat menggunakan banyak buku masak komunitas tanpa modifikasi. Untuk informasi selengkapnya, lihat Menggunakan Pencarian Chef dan Menggunakan Tas Data.
- Sebagian besar konfigurasi tumpukan dan atribut penerapan dijelaskan dalam AWS OpsWorks Referensi Tas Data Stacks dan Atribut Buku Masak Bawaan tersedia untuk resep Windows.

  Anda dapat menggunakan pencarian Chef untuk mendapatkan nilai atribut ini. Sebagai contoh, lihat Memperoleh Nilai Atribut dengan Pencarian Chef. Untuk daftar atribut, lihatAWS OpsWorks Referensi Tas Data Stacks.

Menerapkan Resep untuk Chef 12 Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan Chef 12 memberikan keunggulan berikut dibandingkan tumpukan Chef 11.10:

- Chef menjalankan menggunakan Ruby 2.1.6, sehingga resep Anda dapat menggunakan sintaks Ruby baru.
- Tumpukan Chef 12 dapat menggunakan lebih banyak buku masak komunitas tanpa modifikasi. Tanpa buku masak bawaan, tidak akan ada lagi kemungkinan konflik nama antara buku masak bawaan dan buku masak khusus.

- Anda tidak lagi terbatas pada versi Berkshelf yang AWS OpsWorks Stacks telah menyediakan paket pra-bangun untuk. Berkshelf tidak lagi diinstal pada instance AWS OpsWorks Stacks di Chef 12. Sebagai gantinya, Anda dapat menggunakan versi Berkshelf apa pun di workstation lokal Anda.
- Sekarang ada pemisahan yang jelas antara buku masak bawaan yang disediakan AWS OpsWorks Stacks dengan Chef 12 (Elastic Load Balancing, Amazon RDS, dan Amazon ECS) dan buku masak khusus. Ini membuat pemecahan masalah yang gagal Chef berjalan lebih mudah.

Menerapkan Resep untuk Tumpukan Chef 11.10

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan Chef 11.10 memberikan keunggulan berikut dibandingkan tumpukan Chef 11.4:

- Chef menjalankan menggunakan Ruby 2.0.0, sehingga resep Anda dapat menggunakan sintaks Ruby baru.
- Resep dapat menggunakan pencarian Chef dan tas data.

  Tumpukan Chef 11.10 dapat menggunakan banyak buku masak komunitas tanpa modifikasi.
- Anda dapat menggunakan Berkshelf untuk mengelola buku masak.

  Berkshelf menyediakan cara yang jauh lebih fleksibel untuk mengelola buku masak kustom Anda dan menggunakan buku masak komunitas dalam tumpukan.
- Buku masak harus mendeklarasikan dependensi di. `metadata.rb`

  Jika buku masak Anda bergantung pada buku masak lain, Anda harus memasukkan ketergantungan itu dalam file buku masak Anda. `metadata.rb` Misalnya, jika buku masak Anda menyertakan resep dengan pernyataan seperti`include_recipe anothercookbook::somerecipe`, `metadata.rb` file buku masak Anda harus menyertakan baris berikut: `depends "anothercookbook"`

- AWS OpsWorks Stacks menginstal klien MySQL pada instance stack hanya jika tumpukan menyertakan lapisan MySQL.
- AWS OpsWorks Stacks menginstal klien Ganglia pada instance tumpukan hanya jika tumpukan menyertakan lapisan Ganglia.
- Jika penerapan berjalan `bundle install` dan penginstalan gagal, penerapan juga gagal.

> **Important**
>
> Jangan gunakan kembali nama buku masak bawaan untuk buku masak khusus atau komunitas. Buku masak khusus yang memiliki nama yang sama dengan buku masak bawaan mungkin gagal. Untuk daftar lengkap buku masak bawaan yang tersedia dengan tumpukan Chef 11.10, 11.4, dan 0.9, lihat repositori opsworks-cookbooks di. GitHub
> Buku masak dengan karakter non-ASCII yang berhasil berjalan di tumpukan Chef 0.9 dan 11.4 mungkin gagal pada tumpukan Chef 11.10. Alasannya adalah bahwa tumpukan Chef 11.10 menggunakan Ruby 2.0.0 untuk menjalankan Chef, yang jauh lebih ketat tentang pengkodean daripada Ruby 1.8.7. Untuk memastikan bahwa buku masak tersebut berjalan dengan sukses di tumpukan Chef 11.10, setiap file yang menggunakan karakter non-ASCII harus memiliki komentar di bagian atas yang memberikan petunjuk tentang pengkodean. Misalnya, untuk pengkodean UTF-8, komentarnya adalah. `# encoding: UTF-8` Untuk informasi lebih lanjut tentang pengkodean Ruby 2.0.0, lihat Pengkodean.

Topik

- Instalasi dan Prioritas Buku Masak
- Menggunakan Pencarian Chef
- Menggunakan Tas Data
- Menggunakan Berkshelf

Instalasi dan Prioritas Buku Masak

Prosedur untuk menginstal buku masak AWS OpsWorks Stacks bekerja agak berbeda untuk tumpukan Chef 11.10 daripada versi Chef sebelumnya. Untuk tumpukan Chef 11.10, setelah AWS OpsWorks Stacks menginstal buku masak bawaan, kustom, dan Berkshelf, ia menggabungkannya ke direktori umum dalam urutan berikut:

1. Buku masak bawaan.

2. Buku masak Berkshelf, jika ada.
3. Buku masak khusus, jika ada.

Ketika AWS OpsWorks Stacks melakukan penggabungan ini, ia menyalin seluruh konten direktori, termasuk resep. Jika ada duplikat, aturan berikut berlaku:

- Isi buku masak Berkshelf lebih diutamakan daripada buku masak bawaan.
- Isi buku masak khusus lebih diutamakan daripada buku masak Berkshelf.

Untuk mengilustrasikan bagaimana proses ini bekerja, pertimbangkan skenario berikut, di mana ketiga direktori buku masak menyertakan buku masak bernama: `mycookbook`

- Buku masak bawaan - `mycookbook` termasuk file atribut bernama`someattributes.rb`, file template bernama`sometemplate.erb`, dan resep bernama`somerecipe.rb`.
- Buku masak Berkshelf — `mycookbook` termasuk `sometemplate.erb` dan. `somerecipe.rb`
- Buku masak khusus — `mycookbook` termasuk`somerecipe.rb`.

Buku masak gabungan berisi yang berikut:

- `someattributes.rb`dari buku masak bawaan.
- `sometemplate.erb`dari buku masak Berkshelf.
- `somerecipe.rb`dari buku masak khusus.

> **Important**
>
> Anda tidak boleh menyesuaikan tumpukan Chef 11.10 Anda dengan menyalin seluruh buku masak bawaan ke repositori Anda dan kemudian memodifikasi bagian buku masak. Melakukannya mengesampingkan seluruh buku masak bawaan, termasuk resep. Jika AWS OpsWorks Stacks memperbarui buku masak itu, tumpukan Anda tidak akan mendapatkan manfaat dari pembaruan tersebut kecuali Anda memperbarui salinan pribadi Anda secara manual. Untuk informasi selengkapnya tentang cara menyesuaikan tumpukan, lihatMenyesuaikan Tumpukan AWS OpsWorks.

Menggunakan Pencarian Chef

Anda dapat menggunakan `searchMetode` Chef dalam resep Anda untuk meminta data tumpukan. Anda menggunakan sintaks yang sama seperti yang Anda lakukan untuk server Chef, tetapi AWS OpsWorks Stacks memperoleh data dari objek node lokal alih-alih menanyakan server Chef. Data ini meliputi:

- Konfigurasi tumpukan instans dan atribut penerapan.
- Atribut dari file atribut buku masak bawaan dan kustom instance.
- Data sistem yang dikumpulkan oleh Ohai.

Konfigurasi tumpukan dan atribut penyebaran berisi sebagian besar informasi yang biasanya diperoleh resep melalui pencarian, termasuk data seperti nama host dan alamat IP untuk setiap instance online di tumpukan. AWS OpsWorks Stacks memperbarui atribut ini untuk setiap peristiwa siklus hidup, yang memastikan bahwa atribut tersebut secara akurat mencerminkan status tumpukan saat ini. Ini berarti Anda sering dapat menggunakan resep komunitas yang bergantung pada pencarian di tumpukan Anda tanpa modifikasi. Metode pencarian masih mengembalikan data yang sesuai; itu hanya berasal dari konfigurasi tumpukan dan atribut penerapan alih-alih server.

Keterbatasan utama pencarian AWS OpsWorks Stacks adalah yang menangani hanya data di objek node lokal, konfigurasi tumpukan dan atribut penerapan pada khususnya. Karena alasan itu, jenis data berikut mungkin tidak tersedia melalui penelusuran:

- Atribut yang didefinisikan secara lokal pada instance lain.

  Jika resep mendefinisikan atribut secara lokal, informasi tersebut tidak dilaporkan kembali ke layanan AWS OpsWorks Stacks, sehingga Anda tidak dapat mengakses data tersebut dari instance lain dengan menggunakan penelusuran.

- `deploy`Atribut khusus.

  Anda dapat menentukan JSON kustom saat menerapkan aplikasi dan atribut terkait diinstal pada instance tumpukan untuk penerapan tersebut. Namun, jika Anda menerapkan hanya ke instance yang dipilih, atribut diinstal hanya pada instance tersebut. Kueri untuk atribut JSON kustom tersebut akan gagal pada semua instance lainnya. Selain itu, atribut kustom disertakan dalam konfigurasi tumpukan dan penerapan JSON hanya untuk penerapan tertentu. Mereka hanya dapat diakses sampai peristiwa siklus hidup berikutnya menginstal satu set konfigurasi tumpukan dan atribut penerapan baru. Perhatikan bahwa jika Anda menentukan JSON kustom untuk tumpukan,

atribut diinstal pada setiap instance untuk setiap peristiwa siklus hidup dan selalu dapat diakses melalui penelusuran.

- Data Ohai dari contoh lain.

  Alat Ohai Chef memperoleh berbagai data sistem pada sebuah instance dan menambahkannya ke objek node. Data ini disimpan secara lokal dan tidak dilaporkan kembali ke layanan AWS OpsWorks Stacks, sehingga pencarian tidak dapat mengakses data Ohai dari instance lain. Namun, beberapa data ini mungkin disertakan dalam konfigurasi tumpukan dan atribut penerapan.

- Instans offline.

  Konfigurasi tumpukan dan atribut penerapan berisi data hanya untuk instance online.

Kutipan resep berikut menunjukkan cara mendapatkan alamat IP pribadi dari contoh lapisan PHP dengan menggunakan pencarian.

```
appserver = search(:node, "role:php-app").first
Chef::Log.info("The private IP is '#{appserver[:private_ip]}'")
```

> **Note**
>
> Ketika AWS OpsWorks Stacks menambahkan konfigurasi tumpukan dan atribut penyebaran ke objek node, itu sebenarnya menciptakan dua set atribut lapisan, masing-masing dengan data yang sama. Satu set ada di `layers` namespace, yang merupakan cara AWS OpsWorks Stacks menyimpan data. Set lainnya ada di `role` namespace, yang merupakan cara server Chef menyimpan data yang setara. Tujuan dari `role` namespace adalah untuk memungkinkan kode pencarian yang diimplementasikan untuk server Chef berjalan pada instance AWS OpsWorks Stacks. Jika Anda menulis kode khusus untuk AWS OpsWorks Stacks, Anda dapat menggunakan salah satu `layers:php-app` atau `role:php-app` dalam contoh sebelumnya dan `search` akan mengembalikan hasil yang sama.

Menggunakan Tas Data

Anda dapat menggunakan `data_bag_item`metode Chef dalam resep Anda untuk meminta informasi dalam tas data. Anda menggunakan sintaks yang sama seperti yang Anda lakukan untuk server Chef, tetapi AWS OpsWorks Stacks memperoleh data dari konfigurasi tumpukan instans dan

atribut penerapan. Namun, AWS OpsWorks Stacks saat ini tidak mendukung lingkungan Chef, jadi `node.chef_environment` selalu kembali`_default`.

Anda membuat tas data dengan menggunakan JSON kustom untuk menambahkan satu atau beberapa atribut ke `[:opsworks][:data_bags]` atribut. Contoh berikut menunjukkan format umum untuk membuat tas data di JSON kustom.

> **Note**
>
> Anda tidak dapat membuat tas data dengan menambahkannya ke repositori buku masak Anda. Anda harus menggunakan JSON khusus.

```
{
  "opsworks": {
    "data_bags": {
      "bag_name1": {
        "item_name1: {
          "key1" : “value1”,
          "key2" : “value2”,
          ...
        }
      },
      "bag_name2": {
        "item_name1": {
          "key1" : “value1”,
          "key2" : “value2”,
          ...
        }
      },
      ...
    }
  }
}
```

Anda biasanya menentukan JSON kustom untuk tumpukan, yang menginstal atribut kustom pada setiap instance untuk setiap peristiwa siklus hidup berikutnya. Anda juga dapat menentukan JSON kustom saat menerapkan aplikasi, tetapi atribut tersebut hanya diinstal untuk penerapan tersebut,

dan mungkin diinstal hanya untuk satu set instance yang dipilih. Untuk informasi selengkapnya, lihat Menerapkan Aplikasi.

Contoh JSON kustom berikut membuat tas data bernama`myapp`. Ini memiliki satu item,`mysql`, dengan dua pasangan kunci-nilai.

```
{ "opsworks": {
    "data_bags": {
      "myapp": {
        "mysql": {
          "username": "default-user",
          "password": "default-pass"
        }
      }
    }
  }
}
```

Untuk menggunakan data dalam resep Anda, Anda dapat memanggil `data_bag_item` dan meneruskannya tas data dan nama nilai, seperti yang ditunjukkan pada kutipan berikut.

```
mything = data_bag_item("myapp", "mysql")
Chef::Log.info("The username is '#{mything['username']}' ")
```

Untuk memodifikasi data dalam kantong data, cukup ubah JSON kustom, dan itu akan diinstal pada instance tumpukan untuk acara siklus hidup berikutnya.

Menggunakan Berkshelf

Dengan tumpukan Chef 0.9 dan Chef 11.4, Anda hanya dapat menginstal satu repositori buku masak khusus. Dengan tumpukan Chef 11.10, Anda dapat menggunakan Berkshelf untuk mengelola buku masak dan dependensinya, yang memungkinkan Anda menginstal buku masak dari beberapa repositori. (Untuk informasi selengkapnya, lihat Ketergantungan Buku Masak Kemasan Secara Lokal.) Secara khusus, dengan Berkshelf, Anda dapat menginstal buku masak komunitas yang AWS OpsWorks kompatibel dengan Stacks langsung dari repositori mereka alih-alih harus menyalinnya ke repositori buku masak khusus Anda. Versi Berkshelf yang didukung tergantung pada sistem operasi. Untuk informasi selengkapnya, lihat AWS OpsWorks Stacks sistem operasi.

Untuk menggunakan Berkshelf, Anda harus secara eksplisit mengaktifkannya, seperti yang dijelaskan dalam. Memasang Buku Masak Kustom Kemudian, sertakan `Berksfile` file di direktori root repositori buku masak Anda yang menentukan buku masak mana yang akan diinstal.

Untuk menentukan sumber buku masak eksternal di Berksfile, sertakan atribut sumber di bagian atas file yang menentukan URL repositori default. Berkshelf akan mencari buku masak di sumber URLs kecuali Anda secara eksplisit menentukan repositori. Kemudian sertakan baris untuk setiap buku masak yang ingin Anda instal dalam format berikut:

```
cookbook 'cookbook_name', ['>= cookbook_version'], [cookbook_options]
```

Bidang berikut `cookbook` menentukan buku masak tertentu.

- *`cookbook_name`*— (Wajib) Menentukan nama buku masak ini.

  Jika Anda tidak menyertakan bidang lain, Berkshelf menginstal buku masak dari sumber yang ditentukan. URLs
- *`cookbook_version`*— (Opsional) Menentukan versi buku masak atau versi.

  Anda dapat menggunakan awalan seperti = atau >= untuk menentukan versi tertentu atau rentang versi yang dapat diterima. Jika Anda tidak menentukan versinya, Berkshelf menginstal yang terbaru.
- *`cookbook_options`*— (Opsional) Bidang terakhir adalah hash yang berisi satu atau lebih pasangan kunci-nilai yang menentukan opsi seperti lokasi repositori.

  Misalnya, Anda dapat menyertakan `git` kunci untuk menunjuk repositori Git tertentu dan `tag` kunci untuk menunjuk cabang repositori tertentu. Menentukan cabang repositori biasanya merupakan cara terbaik untuk memastikan bahwa Anda menginstal buku masak pilihan Anda.

> **Important**
>
> Jangan mendeklarasikan buku masak dengan menyertakan `metadata` baris di Berksfile Anda dan mendeklarasikan dependensi buku masak di. `metadata.rb` Agar ini berfungsi dengan benar, kedua file harus berada di direktori yang sama. Dengan AWS OpsWorks Stacks, Berksfile harus berada di direktori root repositori, tetapi `metadata.rb` file

harus berada di direktori buku masak masing-masing. Anda seharusnya secara eksplisit mendeklarasikan buku masak eksternal di Berksfile.

Berikut ini adalah contoh Berksfile yang menunjukkan cara berbeda untuk menentukan buku masak Untuk informasi selengkapnya tentang cara membuat Berksfile, lihat 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'
```

File ini menginstal buku masak berikut:

- Versi terbaru `apt` dari repositori buku masak komunitas.
- Versi terbaru `bluepill` dari buku masak komunitas, asalkan versi 2.3.1 atau yang lebih baru.
- Versi terbaru `ark` dari dari repositori tertentu.

  URL untuk contoh ini adalah untuk repositori buku masak komunitas publik GitHub, tetapi Anda dapat menginstal buku masak dari repositori lain, termasuk repositori pribadi. Untuk informasi selengkapnya, lihat Berkshelf.
- `build-essential`Buku masak dari cabang v1.4.2 dari repositori yang ditentukan.

Repositori buku masak khusus dapat berisi buku masak khusus selain Berksfile. Dalam hal ini, AWS OpsWorks Stacks menginstal kedua set buku masak, yang berarti bahwa sebuah instance dapat memiliki sebanyak tiga repositori buku masak.

- Buku masak bawaan dipasang ke`/opt/aws/opsworks/current/cookbooks`.
- Jika repositori buku masak kustom Anda berisi buku masak, mereka diinstal ke. `/opt/aws/opsworks/current/site-cookbooks`
- Jika Anda telah mengaktifkan Berkshelf dan repositori buku masak kustom Anda berisi Berksfile, buku masak yang ditentukan diinstal ke. `/opt/aws/opsworks/current/berkshelf-cookbooks`

Buku masak bawaan dan buku masak khusus Anda diinstal pada setiap instance selama penyiapan dan tidak diperbarui selanjutnya kecuali Anda menjalankan perintah tumpukan Perbarui Buku Masak Kustom secara manual. AWS OpsWorks Tumpukan berjalan `berks install` untuk setiap Chef run, sehingga buku masak Berkshelf Anda diperbarui untuk setiap acara siklus hidup, sesuai dengan aturan berikut:

- Jika Anda memiliki versi buku masak baru di repositori, operasi ini memperbarui buku masak dari repositori.
- Jika tidak, operasi ini memperbarui buku masak Berkshelf dari cache lokal.

> **Note**
>
> Operasi menimpa buku masak Berkshelf, jadi jika Anda telah memodifikasi salinan lokal dari buku masak apa pun, perubahan akan ditimpa. Untuk informasi lebih lanjut, lihat Berkshelf

Anda juga dapat memperbarui buku masak Berkshelf Anda dengan menjalankan perintah tumpukan Update Custom Cookbooks, yang memperbarui buku masak Berkshelf dan buku masak khusus Anda.

Menerapkan Resep untuk Chef 11.4 Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Important**
>
> Jangan gunakan kembali nama buku masak bawaan untuk buku masak khusus atau komunitas. Buku masak khusus yang memiliki nama yang sama dengan buku masak bawaan mungkin gagal. Untuk daftar lengkap buku masak bawaan yang tersedia dengan tumpukan Chef 11.10, 11.4, dan 0.9, lihat repositori opsworks-cookbooks di. GitHub

Keterbatasan utama tumpukan Chef 11.4 adalah bahwa resep tidak dapat menggunakan pencarian Chef atau kantong data. Namun, AWS OpsWorks Stacks menginstal konfigurasi tumpukan dan atribut penerapan pada setiap instance yang berisi banyak informasi yang akan Anda peroleh dengan pencarian, termasuk yang berikut ini:

- Data yang ditentukan pengguna dari konsol seperti nama host atau aplikasi.
- Tumpuk data konfigurasi yang dihasilkan oleh layanan AWS OpsWorks Stacks, seperti lapisan tumpukan, aplikasi, dan instance, dan detail tentang setiap instance seperti alamat IP.
- Atribut JSON kustom yang berisi data yang disediakan oleh pengguna dan dapat melayani banyak tujuan yang sama seperti tas data.

AWS OpsWorks Stacks menginstal versi konfigurasi tumpukan dan atribut penerapan saat ini pada setiap instance untuk setiap peristiwa siklus hidup, sebelum memulai Chef run acara. Data tersedia untuk resep melalui `node[:attribute][:child_attribute][...]` sintaks standar. Misalnya, konfigurasi tumpukan dan atribut penerapan menyertakan nama tumpukan,`node[:opsworks][:stack][:name]`.

Kutipan berikut dari salah satu resep bawaan memperoleh nama tumpukan dan menggunakannya untuk membuat file konfigurasi.

```
template '/etc/ganglia/gmetad.conf' do
  source 'gmetad.conf.erb'
  mode '0644'
  variables :stack_name => node[:opsworks][:stack][:name]
  notifies :restart, "service[gmetad]"
end
```

Banyak konfigurasi stack dan nilai atribut deployment berisi beberapa atribut. Anda harus mengulangi atribut-atribut ini untuk mendapatkan informasi yang Anda butuhkan. Contoh di bawah ini menunjukkan kutipan dari konfigurasi tumpukan dan atribut penerapan, yang direpresentasikan sebagai objek JSON untuk kenyamanan. Ini berisi atribut tingkat atas`deploy`, yang berisi atribut untuk setiap aplikasi tumpukan, dinamai dengan nama pendek aplikasi.

```
{
  ...
  "deploy": {
```

```
    "app1_shortname": {
      "document_root": "app1_root",
      "deploy_to": "deploy_directory",
      "application_type": "php",
      ...
    },
    "app2_shortname": {
      "document_root": "app2_root",
      ...
    }
  },
  ...
}
```

Setiap atribut aplikasi berisi satu set atribut yang menjadi ciri aplikasi. Misalnya, `deploy_to` atribut mewakili direktori penerapan aplikasi. Kutipan berikut menetapkan pengguna, grup, dan jalur untuk setiap direktori penerapan aplikasi.

```
node[:deploy].each do |application, deploy|
  opsworks_deploy_dir do
    user deploy[:user]
    group deploy[:group]
    path deploy[:deploy_to]
  end
  ...
end
```

Untuk informasi selengkapnya tentang konfigurasi tumpukan dan atribut penerapan, lihatMenyesuaikan Tumpukan AWS OpsWorks. Untuk informasi lebih lanjut tentang penerapan direktori, lihat. Menyebarkan Resep

Tumpukan Chef 11.4 tidak mendukung kantong data, tetapi Anda dapat menambahkan data arbitrer ke konfigurasi tumpukan dan atribut penerapan dengan menentukan JSON khusus. Resep Anda kemudian dapat mengakses data dengan menggunakan sintaks simpul Chef standar. Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom.

Jika Anda memerlukan fungsionalitas tas data terenkripsi, salah satu opsi adalah menyimpan atribut sensitif di lokasi yang aman seperti bucket Amazon S3 pribadi. Resep Anda kemudian dapat menggunakan AWS Ruby SDK —yang diinstal pada semua instans AWS OpsWorks Stacks—untuk mengunduh data dari bucket.

> **Note**
>
> Setiap AWS OpsWorks instans memiliki profil instans. Peran IAM terkait menentukan sumber daya AWS mana yang dapat diakses oleh aplikasi yang sedang berjalan pada instans. Agar resep Anda dapat mengakses bucket Amazon S3, kebijakan peran harus menyertakan pernyataan yang mirip dengan berikut ini, yang memberikan izin untuk mengambil file dari bucket tertentu.
>
> ```
> "Action": ["s3:GetObject"],
> "Effect": "Allow",
> "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
> ```
>
> Untuk informasi selengkapnya tentang profil instans, lihatMenentukan Izin untuk Aplikasi yang Berjalan pada instance EC2 .

Migrasi Stack Linux yang Ada ke Versi Chef baru

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menggunakan konsol AWS OpsWorks Stacks, API, atau CLI untuk memigrasikan tumpukan Linux Anda ke versi Chef yang lebih baru. Namun, resep Anda mungkin memerlukan modifikasi agar kompatibel dengan versi yang lebih baru. Saat bersiap untuk memigrasikan tumpukan, pertimbangkan hal berikut.

- Anda tidak dapat mengubah versi tumpukan AWS OpsWorks Stacks dari Chef 11 ke Chef 12 dengan mengedit atau mengkloning tumpukan. Upgrade versi utama Chef tidak dapat dilakukan menggunakan prosedur di bagian ini. Untuk informasi lebih lanjut tentang transisi Chef 11.10 ke Chef 12, lihatMenerapkan Resep: Chef 12.

- Transisi dari satu versi Chef ke versi lainnya melibatkan sejumlah perubahan, beberapa di antaranya melanggar perubahan.

  Untuk informasi lebih lanjut tentang transisi Chef 0.9 ke Chef 11.4, lihat. Bermigrasi ke Versi Chef baru Untuk informasi lebih lanjut tentang transisi Chef 11.4 ke Chef 11.10, lihat. Resep Penerapan: Chef 11.10 Untuk informasi lebih lanjut tentang transisi Chef 11.10 ke Chef 12, lihatMenerapkan Resep: Chef 12.
- Chef run menggunakan versi Ruby yang berbeda pada tumpukan Chef 0.9 dan Chef 11.4 (Ruby 1.8.7), tumpukan Chef 11.10 (Ruby 2.0.0), dan tumpukan Chef 12 (Ruby 2.1.6).

  Untuk informasi selengkapnya, lihat Versi Ruby.
- Tumpukan Chef 11.10 menangani pemasangan buku masak secara berbeda dari tumpukan Chef 0.9 atau Chef 11.4.

  Perbedaan ini dapat menyebabkan masalah saat memigrasikan tumpukan yang menggunakan buku masak khusus ke Chef 11.10. Untuk informasi selengkapnya, lihat Instalasi dan Prioritas Buku Masak.

Berikut ini adalah panduan yang disarankan untuk memigrasikan tumpukan Chef ke versi Chef yang lebih baru:

Untuk memigrasikan tumpukan ke versi Chef yang lebih baru

1. Kloning tumpukan produksi Anda. Pada halaman Clone Stack, klik Lanjutan>> untuk menampilkan bagian Manajemen Konfigurasi, dan ubah versi Chef ke versi berikutnya yang lebih tinggi.

   > **Note**
   >
   > Jika Anda memulai dengan tumpukan Chef 0.9, Anda tidak dapat meningkatkan langsung ke Chef 11.10; Anda harus terlebih dahulu meningkatkan ke Chef 11.4. Jika Anda ingin memigrasikan tumpukan Anda ke Chef 11.10 sebelum menguji resep Anda, tunggu 20 menit hingga pembaruan dijalankan, lalu tingkatkan tumpukan dari 11.4 ke 11.10.
2. Tambahkan instance ke lapisan dan uji aplikasi dan buku masak tumpukan kloning pada sistem pengujian atau pementasan. Untuk informasi selengkapnya, lihat Semua tentang Chef... .
3. Saat hasil tes memuaskan, lakukan salah satu hal berikut:

- Jika ini adalah versi Chef yang Anda inginkan, Anda dapat menggunakan tumpukan kloning sebagai tumpukan produksi Anda, atau mengatur ulang versi Chef di tumpukan produksi Anda.
- Jika Anda memigrasikan tumpukan Chef 0.9 ke Chef 11.10 dalam dua tahap, ulangi proses untuk memigrasikan tumpukan dari Chef 11.4 ke Chef 11.10.

> **Note**
>
> Saat menguji resep, Anda dapat menggunakan SSH untuk terhubung ke instance dan kemudian menggunakan perintah run_command Instance Agent CLI untuk menjalankan resep yang terkait dengan berbagai peristiwa siklus hidup. Agen CLI sangat berguna untuk menguji resep Pengaturan karena Anda dapat menggunakannya bahkan Pengaturan gagal dan instance tidak mencapai status online. Anda juga dapat menggunakan perintah Setup stack untuk menjalankan kembali resep Setup, tetapi perintah itu hanya tersedia jika Setup berhasil dan instance online.

Dimungkinkan untuk memperbarui tumpukan yang sedang berjalan ke versi Chef baru.

Untuk memperbarui tumpukan yang sedang berjalan ke versi Chef baru

1. Edit tumpukan untuk mengubah pengaturan tumpukan versi Chef.
2. Simpan pengaturan baru dan tunggu AWS OpsWorks Stacks memperbarui instance, yang biasanya memakan waktu 15 - 20 menit.

> **Important**
>
> AWS OpsWorks Tumpukan tidak menyinkronkan pembaruan versi Chef dengan peristiwa siklus hidup. Jika Anda ingin memperbarui versi Chef pada tumpukan produksi, Anda harus berhati-hati untuk memastikan bahwa pembaruan selesai sebelum peristiwa siklus hidup berikutnya terjadi. Jika suatu peristiwa terjadi—biasanya peristiwa Penerapan atau Konfigurasi—agen instans memperbarui buku masak kustom Anda dan menjalankan resep yang ditetapkan acara, baik pembaruan versi selesai atau belum. Tidak ada cara langsung untuk menentukan kapan pembaruan versi selesai, tetapi log penerapan menyertakan versi Chef.

Versi Ruby

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Semua instance dalam tumpukan Linux telah menginstal Ruby. AWS OpsWorks Stacks menginstal paket Ruby pada setiap instance, yang digunakannya untuk menjalankan resep Chef dan agen instance. AWS OpsWorks Stacks menentukan versi Ruby berdasarkan versi Chef mana tumpukan berjalan. Jangan mencoba memodifikasi versi ini; melakukannya mungkin menonaktifkan agen instance.

AWS OpsWorks Stacks tidak menginstal aplikasi Ruby yang dapat dieksekusi di tumpukan Windows. Klien Chef 12.2 hadir dengan Ruby 2.0.0 p451, tetapi executable Ruby tidak ditambahkan ke variabel lingkungan PATH instance. Jika Anda ingin menggunakan executable ini untuk menjalankan kode Ruby, itu terletak `\opscode\chef\embedded\bin\ruby.exe` di drive Windows Anda.

Tabel berikut merangkum versi AWS OpsWorks Tumpukan Ruby. Aplikasi yang tersedia versi Ruby juga tergantung pada sistem operasi instans. Untuk informasi selengkapnya, termasuk versi tambalan yang tersedia, lihatAWS OpsWorks Stacks sistem operasi.

Versi Chef	Versi Chef Ruby	Tersedia Aplikasi Versi Ruby
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	Tidak ada
12.22 (d)	2.3.6	Tidak ada

(a) Tidak tersedia dengan Amazon Linux 2014.09 dan yang lebih baru, Red Hat Enterprise Linux (RHEL), atau Ubuntu 14.04 LTS.

(b) Hanya tersedia di tumpukan Linux.

(c) Tidak tersedia dengan RHEL.

(d) Hanya tersedia di tumpukan Windows. Versi utama adalah 12.2. Versi minor saat ini adalah 12.22.

(e) Penghentian selesai; dukungan telah berakhir.

Lokasi penginstalan bergantung pada versi Chef:

- Aplikasi menggunakan `/usr/local/bin/ruby` executable untuk semua Versi Chef.
- Untuk Chef 0.9 dan 11.4, agen instans dan resep Chef menggunakan executable. `/usr/bin/ruby`
- Untuk Chef 11.10, resep instans dan resep Chef menggunakan resep yang dapat `/opt/aws/opsworks/local/bin/ruby` dieksekusi.

Memasang Buku Masak Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

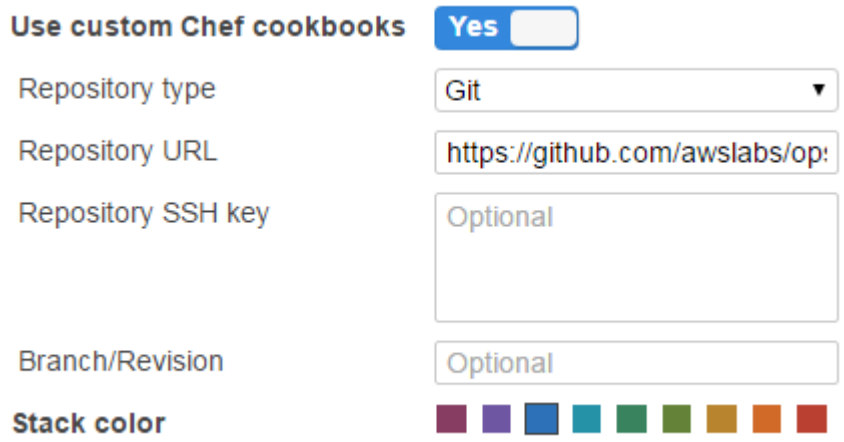
Untuk menginstal tumpukan dan menggunakan buku masak khusus, Anda harus mengonfigurasi tumpukan untuk mengaktifkan buku masak khusus, jika belum dikonfigurasi. Anda kemudian harus memberikan URL repositori dan informasi terkait apa pun seperti kata sandi.

> **Important**
>
> Setelah Anda mengonfigurasi tumpukan untuk mendukung buku masak khusus, AWS OpsWorks Stacks secara otomatis menginstal buku masak Anda di semua instance baru

saat startup. Namun, Anda harus secara eksplisit mengarahkan AWS OpsWorks Stacks untuk menginstal buku masak baru atau yang diperbarui pada instance yang ada dengan menjalankan perintah tumpukan Update Custom Cookbooks. Untuk informasi selengkapnya, lihat Memperbarui Buku Masak Kustom. Sebelum Anda mengaktifkan Gunakan buku masak Chef khusus di tumpukan Anda, pastikan bahwa buku masak khusus dan komunitas yang Anda jalankan mendukung versi Chef yang digunakan tumpukan Anda.

Untuk mengonfigurasi tumpukan untuk buku masak khusus

1. Di halaman tumpukan Anda, klik Pengaturan Tumpukan untuk menampilkan halaman Pengaturannya., Klik Edit untuk mengedit pengaturan.
2. Alihkan Gunakan buku masak Chef khusus ke Ya.



3. Konfigurasikan buku masak khusus Anda.

Setelah selesai, klik Simpan untuk menyimpan tumpukan yang diperbarui.

Mencantumkan Repositori Cookbook Kustom

Tumpukan Linux dapat menginstal buku masak khusus dari salah satu jenis repositori berikut:

- Arsip HTTP atau Amazon S3.

  Mereka dapat berupa publik atau pribadi, tetapi Amazon S3 biasanya merupakan opsi yang lebih disukai untuk arsip pribadi.
- Repositori Git dan Subversion menyediakan kontrol sumber dan kemampuan untuk memiliki beberapa versi.

Tumpukan Windows dapat menginstal buku masak khusus dari arsip Amazon S3 dan repositori Git.

Semua jenis repositori memiliki bidang wajib berikut.

- Jenis repositori —Jenis repositori
- URL Repositori —URL repositori

AWS OpsWorks Stacks mendukung situs repositori Git yang dihosting secara publik seperti GitHubatau Bitbucket serta server Git yang dihosting secara pribadi. Untuk repositori Git, Anda harus menggunakan salah satu format URL berikut, tergantung pada apakah repositori bersifat publik atau pribadi. Ikuti pedoman URL yang sama untuk submodul Git.

Untuk repositori Git publik, gunakan protokol hanya-baca HTTPS atau Git:

- Git hanya-baca —. `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`
- HTTPS —`https://github.com/amazonwebservices/opsworks-example-cookbooks.git`.

Untuk repositori Git privat, Anda harus menggunakan format baca/tulis SSH, seperti yang ditunjukkan pada contoh berikut:

- Repositori Github —. `git@github.com:`*`project`*`/`*`repository`*
- Repositori pada server Git — *`user`*`@`*`server`*`:`*`project`*`/`*`repository`*

Pengaturan yang tersisa bervariasi dengan tipe repositori dan dijelaskan di bagian berikut.

HTTP Arsip

Memilih Arsip Http untuk jenis Repositori menampilkan dua pengaturan tambahan, yang harus Anda selesaikan jika arsip dilindungi kata sandi.

- Nama pengguna —Nama pengguna Anda
- Kata Sandi —Kata Sandi Anda

Arsip Amazon S3

Memilih Arsip S3 untuk jenis Repositori menampilkan pengaturan opsional tambahan berikut. AWS OpsWorks Tumpukan dapat mengakses repositori Anda dengan menggunakan EC2 peran Amazon (otentikasi manajer sistem operasi host), baik Anda menggunakan AWS OpsWorks Stacks API atau konsol.

- ID kunci akses —ID kunci akses AWS, sepertiAKIAIOSFODNN7EXAMPLE.
- Kunci akses rahasia — Kunci akses rahasia AWS yang sesuai, sepertiwJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY.

Git Repositori

Memilih Git di bawah Kontrol Sumber menampilkan pengaturan opsional tambahan berikut:

Kunci SSH repositori

Anda harus menentukan kunci SSH deploy untuk mengakses repositori Git pribadi. Untuk submodul Git, kunci yang ditentukan harus memiliki akses ke submodul tersebut. Untuk informasi selengkapnya, lihat Menggunakan Kunci SSH Repositori Git.

> **Important**
>
> Kunci SSH penerapan tidak dapat memerlukan kata sandi; AWS OpsWorks Tumpukan tidak memiliki cara untuk melewatinya.

Cabang/Revisi

Jika repositori memiliki beberapa cabang, AWS OpsWorks Stacks mengunduh cabang master secara default. Untuk menentukan cabang tertentu, masukkan nama cabang, SHA1 hash, atau nama tag. Untuk menentukan komit tertentu, masukkan ID komit 40 hekdigit penuh.

Repositori Subversi

Memilih Subversion di bawah Source Control menampilkan pengaturan tambahan berikut:

- Nama pengguna —Nama pengguna Anda, untuk repositori pribadi.
- Kata sandi —Kata sandi Anda, untuk repositori pribadi.

- Revisi — [Opsional] Nama revisi, jika Anda memiliki beberapa revisi.

  Untuk menentukan cabang atau tag, Anda harus memodifikasi URL repositori, misalnya: **`http://repository_domain/repos/myapp/branches/my-apps-branch`** atau. **`http://repository_domain_name/repos/calc/myapp/my-apps-tag`**

Memperbarui Buku Masak Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Saat Anda menyediakan AWS OpsWorks Stacks dengan buku masak khusus, resep Pengaturan bawaan membuat cache lokal pada setiap instance yang baru dimulai, dan mengunduh buku masak ke cache. AWS OpsWorks Tumpukan kemudian menjalankan resep dari cache, bukan repositori. Jika Anda memodifikasi buku masak kustom di repositori, Anda harus memastikan bahwa buku masak yang diperbarui diinstal pada cache lokal instans Anda. AWS OpsWorks Stacks secara otomatis menyebarkan buku masak terbaru ke instance baru saat dimulai. Namun, untuk kasus yang ada, situasinya berbeda:

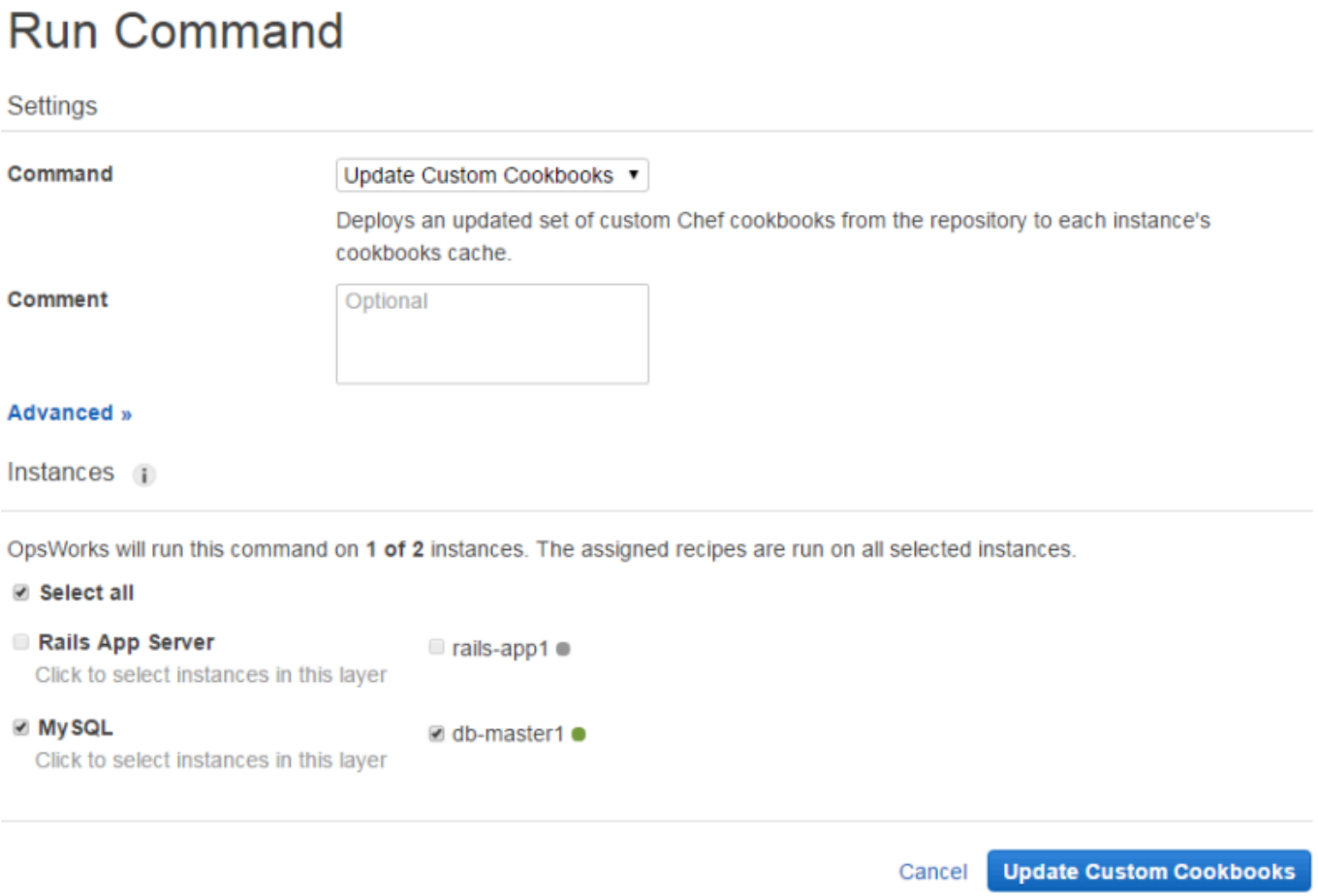
- Anda harus menyebarkan buku masak kustom yang diperbarui secara manual ke instance online.
- Anda tidak perlu menerapkan buku masak kustom yang diperbarui ke instans yang didukung toko instans offline, termasuk instans berbasis beban dan berbasis waktu.

  AWS OpsWorks Stacks secara otomatis menyebarkan buku masak saat ini ketika instance dimulai ulang.
- Anda harus memulai instance 24/7 yang didukung EBS offline yang tidak berbasis beban atau berbasis waktu.
- Anda tidak dapat memulai instans berbasis beban dan berbasis waktu yang didukung EBS offline, jadi pendekatan paling sederhana adalah menghapus instance offline dan menambahkan instance baru untuk menggantikannya.

Karena mereka sekarang adalah instance baru, AWS OpsWorks Stacks secara otomatis menyebarkan buku masak kustom saat ini ketika instance dimulai.

Untuk memperbarui buku masak kustom secara manual

1. Perbarui repositori Anda dengan buku masak yang dimodifikasi. AWS OpsWorks Stacks menggunakan URL cache yang Anda berikan ketika Anda awalnya menginstal buku masak, sehingga nama file root buku masak, lokasi repositori, dan hak akses tidak boleh berubah.
   - Untuk repositori Amazon S3 atau HTTP, ganti file.zip asli dengan file.zip baru yang memiliki nama yang sama.
   - Untuk repositori Git atau Subversion, edit pengaturan tumpukan Anda untuk mengubah bidang Cabang/Revisi ke versi baru.
2. Pada halaman stack, klik Run Command dan pilih perintah Update Custom Cookbooks.



3. Tambahkan komentar jika diinginkan.

4. Secara opsional, tentukan objek JSON khusus untuk perintah untuk menambahkan atribut khusus ke konfigurasi tumpukan dan atribut penerapan yang diinstal AWS OpsWorks Stacks pada instance. Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom dan Mengesampingkan Atribut.
5. Secara default, AWS OpsWorks Stacks memperbarui buku masak pada setiap instance. Untuk menentukan instance mana yang akan diperbarui, pilih instance yang sesuai dari daftar di akhir halaman. Untuk memilih setiap instance dalam layer, pilih kotak centang layer yang sesuai di kolom kiri.
6. Klik Perbarui Buku Masak Kustom untuk menginstal buku masak yang diperbarui. AWS OpsWorks Stacks menghapus buku masak khusus yang di-cache pada instance yang ditentukan dan menginstal buku masak baru dari repositori.

> **Note**
>
> Prosedur ini hanya diperlukan untuk contoh yang ada, yang memiliki versi lama buku masak di cache mereka. Jika Anda kemudian menambahkan instance ke lapisan, AWS OpsWorks Stacks menyebarkan buku masak yang saat ini ada di repositori sehingga mereka secara otomatis mendapatkan versi terbaru.

Resep Eksekusi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menjalankan resep dengan dua cara:

- Secara otomatis, dengan menetapkan resep ke peristiwa siklus hidup lapisan yang sesuai.
- Secara manual, dengan menjalankan perintah stack Execute Recipes atau dengan menggunakan agen CLI.

Topik

- AWS OpsWorks Acara Siklus Hidup Stacks
- Resep Menjalankan Secara Otomatis
- Resep yang berjalan secara manual

AWS OpsWorks Acara Siklus Hidup Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setiap lapisan memiliki satu set lima peristiwa siklus hidup, yang masing-masing memiliki serangkaian resep terkait yang spesifik untuk lapisan tersebut. Ketika sebuah peristiwa terjadi pada instance layer, AWS OpsWorks Stacks secara otomatis menjalankan set resep yang sesuai. Untuk memberikan respons khusus terhadap peristiwa ini, terapkan resep khusus dan tetapkan ke acara yang sesuai untuk setiap lapisan. AWS OpsWorks Stacks menjalankan resep tersebut setelah resep bawaan acara.

Setup

Peristiwa ini terjadi setelah instance dimulai selesai booting. Anda juga dapat memicu Setup acara secara manual dengan menggunakan perintah Setup stack. AWS OpsWorks Stacks menjalankan resep yang mengatur instance sesuai dengan lapisannya. Misalnya, jika instance adalah anggota layer Rails App Server, Setup resep menginstal Apache, Ruby Enterprise Edition, Passenger dan Ruby on Rails.

> **Note**
>
> Acara Setup mengambil instance di luar layanan. Karena instance tidak dalam status Online saat peristiwa siklus hidup Penyiapan berjalan, instance tempat Anda menjalankan peristiwa Penyiapan akan dihapus dari penyeimbang beban.

Configure

Peristiwa ini terjadi pada semua instans tumpukan ketika salah satu hal berikut terjadi:

- Sebuah instance memasuki atau meninggalkan status online.
- Anda mengaitkan alamat IP Elastis dengan instans atau memisahkan satu dari sebuah instance.
- Anda memasang penyeimbang beban Elastic Load Balancing ke lapisan, atau melepaskannya dari lapisan.

Misalnya, tumpukan Anda memiliki instance A, B, dan C, dan Anda memulai instance baru, D. Setelah D selesai menjalankan resep penyiapannya, AWS OpsWorks Stacks memicu Configure peristiwa pada A, B, C, dan D. Jika Anda kemudian menghentikan A, AWS OpsWorks Stacks memicu Configure peristiwa di B, C, dan D. AWS OpsWorks Stacks merespons Configure acara dengan menjalankan Configure resep setiap lapisan, yang memperbarui konfigurasi instance untuk mencerminkan set saat ini contoh online. Oleh karena itu, Configure acara ini adalah saat yang tepat untuk membuat ulang file konfigurasi. Misalnya, HAProxy Configure resep mengkonfigurasi ulang penyeimbang beban untuk mengakomodasi perubahan apa pun dalam kumpulan instance server aplikasi online.

Anda juga dapat memicu peristiwa Configure secara manual dengan menggunakan perintah Configure stack.

Deploy

Peristiwa ini terjadi ketika Anda menjalankan perintah Deploy, biasanya untuk menyebarkan aplikasi ke serangkaian instance server aplikasi. Instance menjalankan resep yang menyebarkan aplikasi dan file terkait apa pun dari repositorinya ke instance layer. Misalnya, untuk instance Rails Application Server, Deploy resep memeriksa aplikasi Ruby yang ditentukan dan memberi tahu Phusion Passenger untuk memuatnya kembali. Anda juga dapat menjalankan Deploy instans lain sehingga mereka dapat, misalnya, memperbarui konfigurasi mereka untuk mengakomodasi aplikasi yang baru diterapkan.

> **Note**
>
> Pengaturan termasuk Deploy; itu menjalankan resep Deploy setelah penyiapan selesai.

Undeploy

Peristiwa ini terjadi saat Anda menghapus aplikasi atau menjalankan Undeploy perintah untuk menghapus aplikasi dari sekumpulan instance server aplikasi. Instance yang ditentukan menjalankan resep untuk menghapus semua versi aplikasi dan melakukan pembersihan yang diperlukan.

Shutdown

Peristiwa ini terjadi setelah Anda mengarahkan AWS OpsWorks Stacks untuk mematikan instance tetapi sebelum EC2 instans Amazon terkait benar-benar dihentikan. AWS OpsWorks Stacks menjalankan resep untuk melakukan tugas pembersihan seperti mematikan layanan.

Jika Anda telah memasang penyeimbang beban Elastic Load Balancing ke layer dan mengaktifkan dukungan untuk pengurasan koneksi, AWS OpsWorks Stacks menunggu hingga pengurasan koneksi selesai sebelum memicu peristiwa. Shutdown

Setelah memicu Shutdown peristiwa, AWS OpsWorks Stacks mengizinkan Shutdown resep dengan jumlah waktu tertentu untuk melakukan tugasnya, lalu menghentikan atau menghentikan instance Amazon. EC2 Nilai Shutdown batas waktu default adalah 120 detik. Jika Shutdown resep Anda mungkin memerlukan lebih banyak waktu, Anda dapat mengedit konfigurasi lapisan untuk mengubah nilai batas waktu. Untuk informasi lebih lanjut tentang contohShutdown, lihatMenghentikan sebuah Instance.

> **Note**
>
> Mem-boot ulang instance tidak memicu peristiwa siklus hidup apa pun.

Untuk diskusi selengkapnya tentang perintah Deploy dan Undeploy aplikasi, lihatMenerapkan Aplikasi.

Setelah instance start selesai booting, urutan startup yang tersisa adalah sebagai berikut:

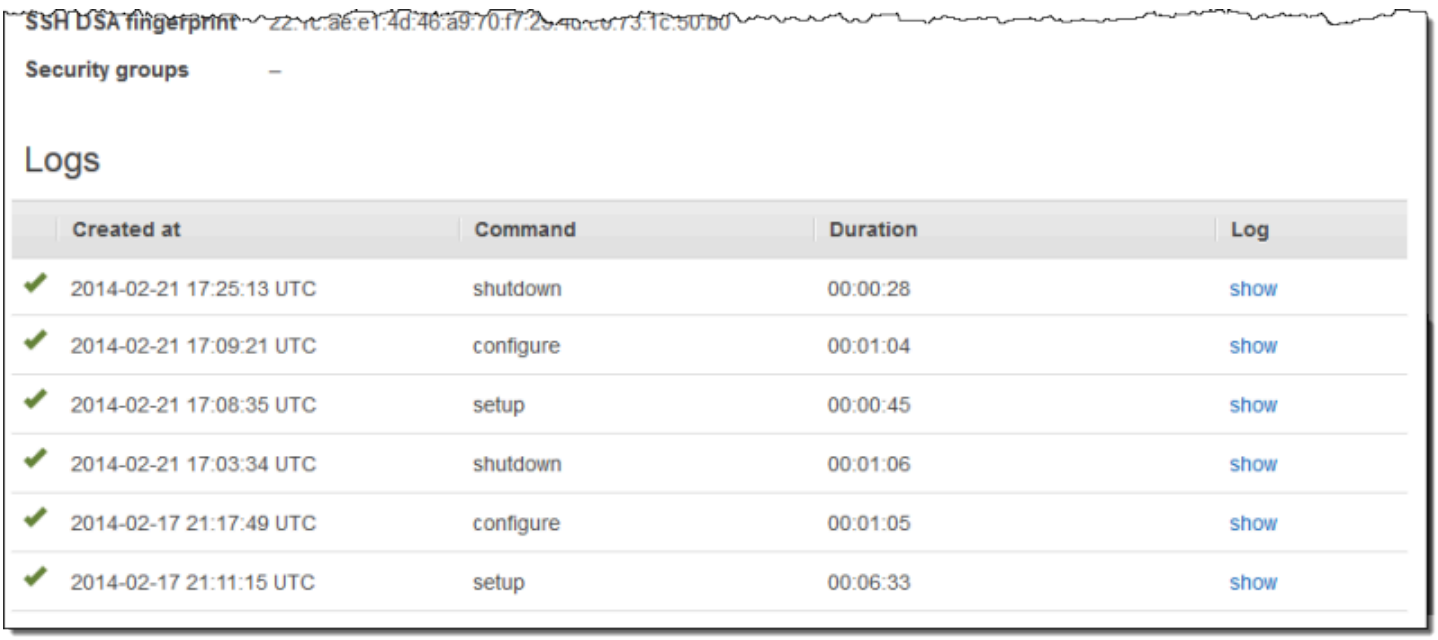
1. AWS OpsWorks Stacks menjalankan Setup resep bawaan instans, diikuti oleh Setup resep khusus apa pun.
2. AWS OpsWorks Stacks menjalankan Deploy resep bawaan instans, diikuti oleh Deploy resep khusus apa pun.

   Contohnya sekarang online.

3. AWS OpsWorks Tumpukan memicu Configure peristiwa pada semua instance di tumpukan, termasuk instance yang baru dimulai.

   AWS OpsWorks Stacks menjalankan Configure resep bawaan instance, diikuti oleh resep khusus apa pun. Configure

> **Note**
>
> Untuk melihat peristiwa siklus hidup yang terjadi pada instance tertentu, buka halaman Instans dan klik nama instans untuk membuka halaman detailnya. Daftar peristiwa ada di bagian Log di bagian bawah halaman. Anda dapat mengklik tampilkan di kolom Log untuk memeriksa log Chef untuk suatu acara. Ini memberikan informasi rinci tentang bagaimana acara itu ditangani, termasuk resep mana yang dijalankan. Untuk informasi lebih lanjut tentang cara menafsirkan log Chef, lihatLog Koki.



Untuk setiap peristiwa siklus hidup, AWS OpsWorks Stacks menginstal satu set konfigurasi tumpukan dan atribut penerapan pada setiap instance yang berisi status tumpukan saat ini dan, untuk Deploy peristiwa, informasi tentang penerapan. Atribut termasuk informasi tentang contoh apa yang tersedia, alamat IP mereka, dan sebagainya. Untuk informasi selengkapnya, lihat Konfigurasi Stack dan Atribut Deployment.

> **Note**
>
> Memulai atau menghentikan sejumlah besar contoh pada saat yang sama dapat dengan cepat menghasilkan sejumlah besar Configure peristiwa. Untuk menghindari pemrosesan yang tidak perlu, AWS OpsWorks Stacks hanya merespons peristiwa terakhir. Konfigurasi tumpukan dan atribut penerapan acara itu berisi semua informasi yang diperlukan untuk memperbarui instance tumpukan untuk seluruh rangkaian perubahan. Hal ini menghilangkan kebutuhan untuk juga memproses Configure peristiwa sebelumnya. AWS OpsWorks Stacks melabeli Configure peristiwa yang belum diproses sebagai digantikan.

Resep Menjalankan Secara Otomatis

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setiap lapisan memiliki satu set resep bawaan yang ditetapkan untuk setiap peristiwa siklus hidup, meskipun beberapa lapisan tidak memiliki resep Undeploy. Ketika peristiwa siklus hidup terjadi pada sebuah instance, AWS OpsWorks Stacks menjalankan kumpulan resep yang sesuai untuk lapisan terkait.

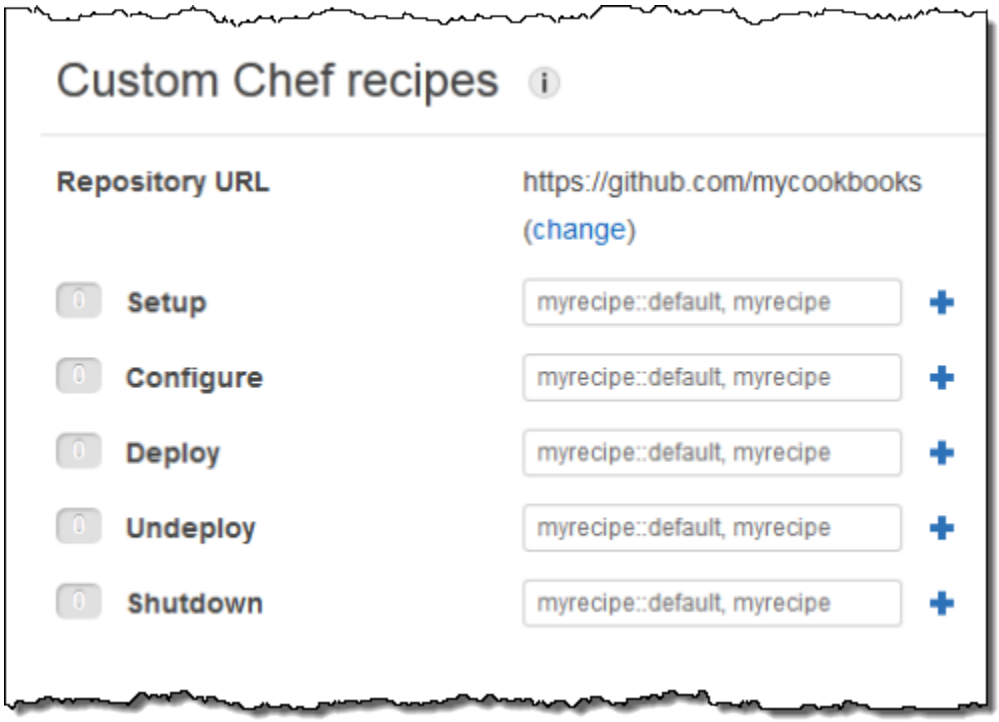
Jika Anda telah menginstal buku masak khusus, Anda dapat membuat AWS OpsWorks Stacks menjalankan beberapa atau semua resep secara otomatis dengan menetapkan setiap resep ke acara siklus hidup lapisan. Setelah peristiwa terjadi, AWS OpsWorks Stacks menjalankan resep kustom yang ditentukan setelah resep bawaan layer.

Untuk menetapkan resep kustom ke acara lapisan

1. Pada halaman Layers, untuk layer yang sesuai, klik Recipes dan kemudian klik Edit. Jika Anda belum mengaktifkan buku masak khusus, klik konfigurasi buku masak untuk membuka halaman Pengaturan tumpukan. Alihkan Gunakan Buku Masak Koki khusus ke Ya, dan berikan informasi

repositori buku masak. Kemudian klik Simpan dan arahkan kembali ke halaman edit untuk tab Resep. Untuk informasi selengkapnya, lihat Memasang Buku Masak Kustom.

2. Pada tab Resep, masukkan setiap resep khusus di bidang acara yang sesuai dan klik + untuk menambahkannya ke daftar. Tentukan resep sebagai berikut*cookbook*::: *somerecipe* (hilangkan `.rb` ekstensi).



Saat Anda memulai instance baru, AWS OpsWorks Stacks secara otomatis menjalankan resep khusus untuk setiap acara, setelah menjalankan resep standar.

> **Note**
>
> Resep khusus dijalankan dalam urutan yang Anda masukkan di konsol. Cara alternatif untuk mengontrol perintah eksekusi adalah dengan menerapkan resep meta yang mengeksekusi resep dalam urutan yang benar.

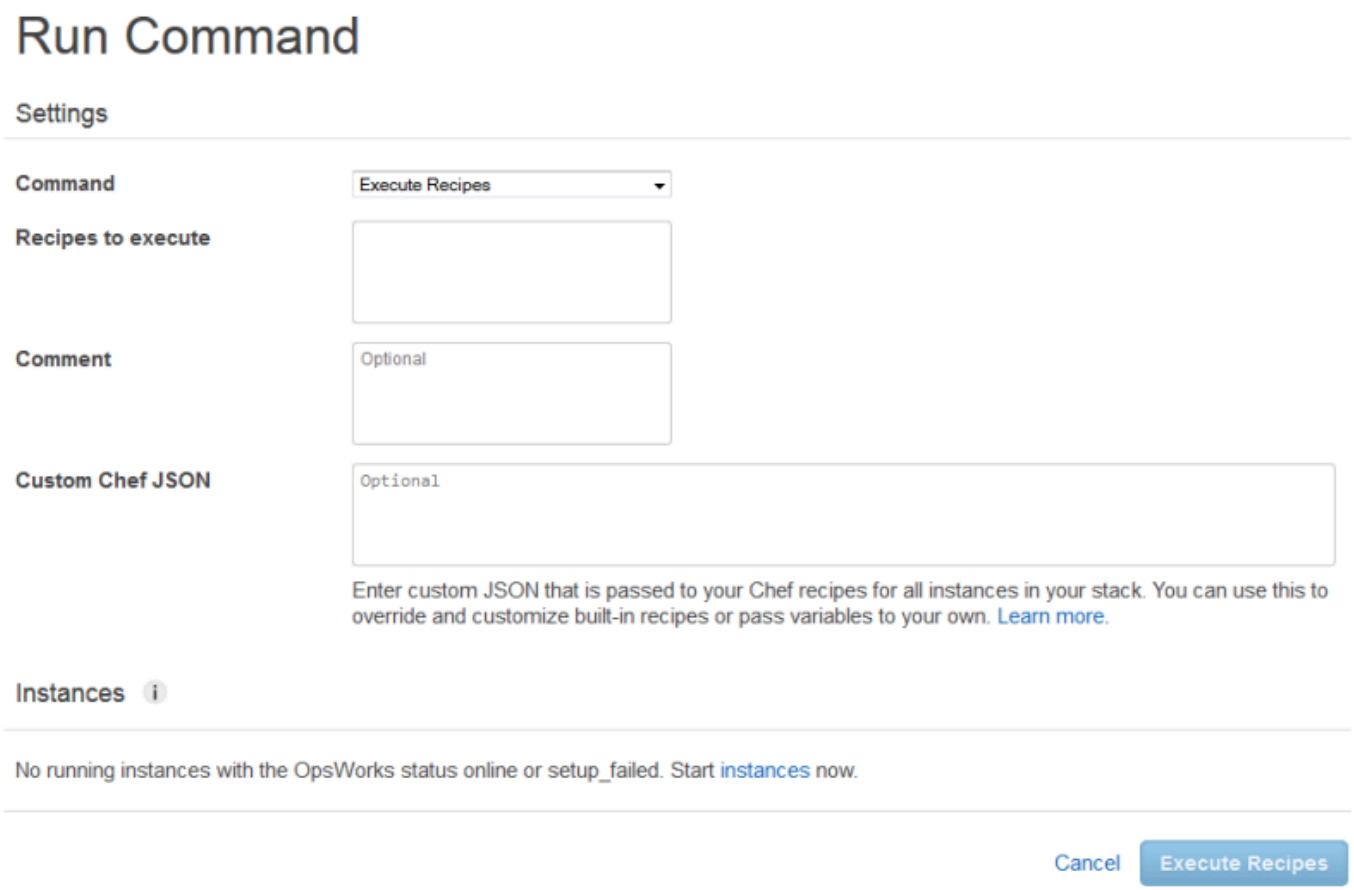
Resep yang berjalan secara manual

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Meskipun resep biasanya dijalankan secara otomatis sebagai respons terhadap peristiwa siklus hidup, Anda dapat menjalankan resep secara manual kapan saja pada salah satu atau semua instance tumpukan. Fitur ini biasanya digunakan untuk tugas yang tidak memetakan dengan baik ke peristiwa siklus hidup, seperti mencadangkan instance. Untuk menjalankan resep khusus secara manual, itu harus ada di salah satu buku masak khusus Anda, tetapi tidak harus ditetapkan ke acara siklus hidup. Saat Anda menjalankan resep secara manual, AWS OpsWorks Stacks menginstal `deploy` atribut yang sama dengan yang dilakukannya untuk acara Deploy.

Untuk menjalankan resep secara manual pada instance tumpukan

1. Pada halaman Stack, klik Jalankan perintah. Untuk Command, pilih Execute Recipes.



2. Masukkan resep yang akan dijalankan di kotak Resep untuk mengeksekusi dengan menggunakan *recipename* format standar*cookbookname*::. Gunakan koma untuk memisahkan beberapa resep; resep tersebut akan berjalan dalam urutan yang Anda daftarkan.
3. Secara opsional, gunakan kotak Custom Chef JSON untuk menambahkan objek JSON kustom yang mendefinisikan atribut kustom yang akan digabungkan ke dalam konfigurasi tumpukan dan atribut penerapan yang diinstal pada instance. Untuk informasi selengkapnya tentang penggunaan objek JSON kustom, lihat Menggunakan JSON Kustom danMengesampingkan Atribut.
4. Di bawah Instance, pilih instance di mana AWS OpsWorks Stacks harus menjalankan resep.

Ketika peristiwa siklus hidup terjadi, agen AWS OpsWorks Stacks menerima perintah untuk menjalankan resep terkait. Anda dapat menjalankan perintah ini secara manual pada instance tertentu dengan menggunakan perintah stack yang sesuai atau dengan menggunakan run_command perintah agen CLI. Untuk informasi lebih lanjut tentang cara menggunakan agen CLI, lihat. AWS OpsWorks Agen Tumpukan CLI

Manajemen Sumber Daya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Halaman Sumber Daya memungkinkan Anda menggunakan alamat IP Elastis akun, volume Amazon EBS, atau sumber daya instans Amazon RDS di AWS OpsWorks tumpukan Stacks. Anda dapat menggunakan Sumber Daya untuk melakukan hal berikut:

- Daftarkan sumber daya dengan tumpukan, yang memungkinkan Anda melampirkan sumber daya ke salah satu instance tumpukan.
- Lampirkan sumber daya ke salah satu instance tumpukan.
- Pindahkan sumber daya dari satu contoh ke contoh lainnya.
- Lepaskan sumber daya dari sebuah instance. Sumber daya tetap terdaftar dan dapat dilampirkan ke contoh lain.
- Deregister sumber daya. Sumber daya yang tidak terdaftar tidak dapat digunakan oleh AWS OpsWorks Stacks, tetapi tetap ada di akun Anda kecuali Anda menghapusnya, dan dapat didaftarkan dengan tumpukan lain.

Perhatikan kendala berikut:

- Anda tidak dapat melampirkan volume Amazon EBS terdaftar ke instans Windows.
- Halaman Sumber Daya mengelola volume Amazon EBS standar, PIOPS, Throughput Optimized HDD, Cold HDD, atau General Purpose (SSD) Amazon EBS, tetapi bukan array RAID.
- Volume Amazon EBS harus diformat xfs.

  AWS OpsWorks Stacks tidak mendukung format file lain, seperti ext4. Untuk informasi selengkapnya tentang menyiapkan volume Amazon EBS, lihat Membuat Volume Amazon EBS Tersedia untuk Digunakan.

- Anda tidak dapat melampirkan volume Amazon EBS ke—atau melepaskannya dari—instance yang sedang berjalan.

  Anda hanya dapat beroperasi pada instance offline. Misalnya, Anda dapat mendaftarkan volume yang sedang digunakan dengan tumpukan dan melampirkannya ke instance offline, tetapi Anda harus menghentikan instance asli dan melepaskan volume sebelum memulai instance baru. Jika tidak, proses awal akan gagal.
- Semua sumber daya terdaftar dikelola hanya di AWS OpsWorks. Ini dapat mengganti properti siklus hidup sumber daya, seperti `DeleteOnTermination` untuk volume. EC2
- Anda dapat melampirkan alamat IP Elastis ke dan melepaskannya dari instance yang sedang berjalan.

  Anda dapat beroperasi pada instans online atau offline. Misalnya, Anda dapat mendaftarkan alamat yang sedang digunakan dan menetapkannya ke instance yang sedang berjalan, dan AWS OpsWorks Stacks akan secara otomatis menetapkan kembali alamat tersebut.
- Untuk mendaftarkan dan membatalkan pendaftaran sumber daya, kebijakan IAM Anda harus memberikan izin untuk tindakan berikut:

Volume Amazon EBS	Alamat IP Elastis	Instans Amazon RDS
RegisterVolume	RegisterElasticIp	RegisterRdsDbInstance
UpdateVolume	UpdateElasticIp	UpdateRdsDbInstance
DeregisterVolume	DeregisterElasticIp	DeregisterRdsDbInstance

Tingkat Kelola izin memberikan izin untuk semua tindakan ini. Untuk mencegah pengguna Kelola mendaftarkan atau membatalkan pendaftaran sumber daya tertentu, edit kebijakan IAM mereka untuk menolak izin untuk tindakan yang sesuai. Untuk informasi selengkapnya, lihat Keamanan dan Izin.

Topik

- Mendaftarkan Sumber Daya dengan Stack
- Melampirkan dan Memindahkan Sumber Daya
- Melepaskan Sumber Daya

- Sumber Daya Deregistering

Mendaftarkan Sumber Daya dengan Stack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Volume Amazon EBS atau alamat IP Elastis harus terdaftar dengan tumpukan sebelum Anda dapat melampirkannya ke instance. Ketika AWS OpsWorks Stacks membuat sumber daya untuk tumpukan, mereka secara otomatis terdaftar dengan tumpukan itu. Jika Anda ingin menggunakan sumber daya yang dibuat secara eksternal, Anda harus mendaftarkannya secara eksplisit. Perhatikan hal berikut:

- Anda dapat mendaftarkan sumber daya hanya dengan satu tumpukan pada satu waktu.
- Saat Anda menghapus tumpukan, AWS OpsWorks Stacks membatalkan pendaftaran semua sumber daya.

Topik

- Mendaftarkan Volume Amazon EBS dengan Stack
- Mendaftarkan Alamat IP Elastis dengan Stack
- Mendaftarkan Instans Amazon RDS dengan Stack

Mendaftarkan Volume Amazon EBS dengan Stack

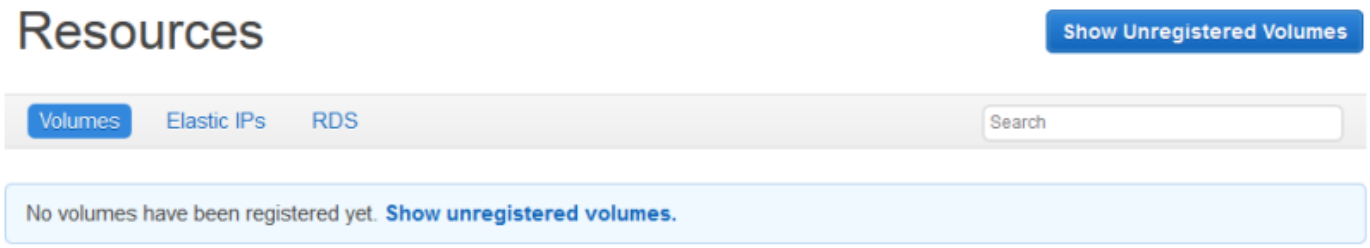
> **Note**
>
> Sumber daya ini hanya dapat digunakan dengan tumpukan Linux. Meskipun Anda dapat mendaftarkan volume Amazon EBS dengan tumpukan Windows, Anda tidak dapat melampirkannya ke instance.

Anda dapat menggunakan halaman Sumber Daya untuk mendaftarkan volume Amazon EBS dengan tumpukan, tunduk pada batasan berikut:

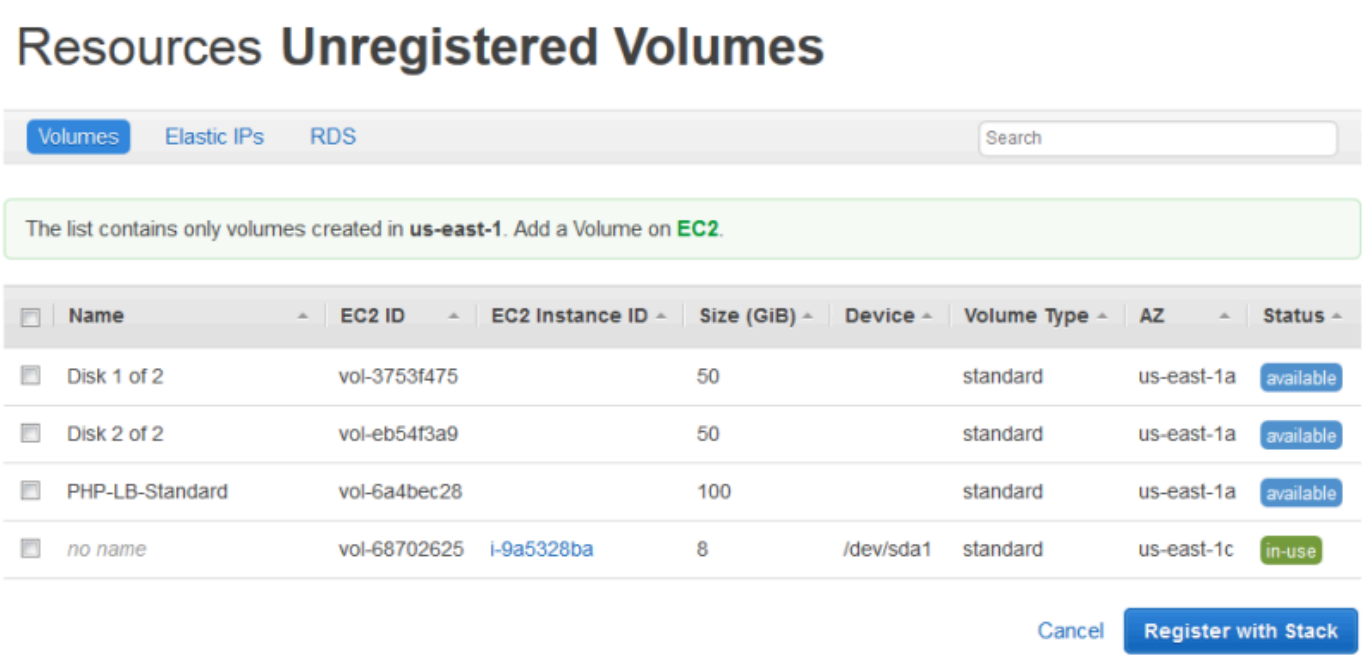
- Volume Amazon EBS non-root yang terpasang harus standar, Throughput Optimized HDD, Cold HDD, PIOPS, atau General Purpose (SSD), tetapi bukan array RAID. Untuk informasi tentang ukuran volume maksimum dan minimum, lihat Volume EBS di panduan ini.
- Volume harus diformat XFS.
- AWS OpsWorks Stacks tidak mendukung format file lain, seperti sistem file diperpanjang keempat (ext4), untuk volume Amazon EBS non-root. Untuk informasi selengkapnya tentang menyiapkan volume Amazon EBS, lihat Membuat Volume Amazon EBS Tersedia untuk Digunakan. Perhatikan bahwa contoh dalam topik tersebut menjelaskan cara membuat volume berbasis ext4, tetapi Anda dapat mengikuti langkah yang sama untuk volume berbasis XFS.

Untuk mendaftarkan volume Amazon EBS

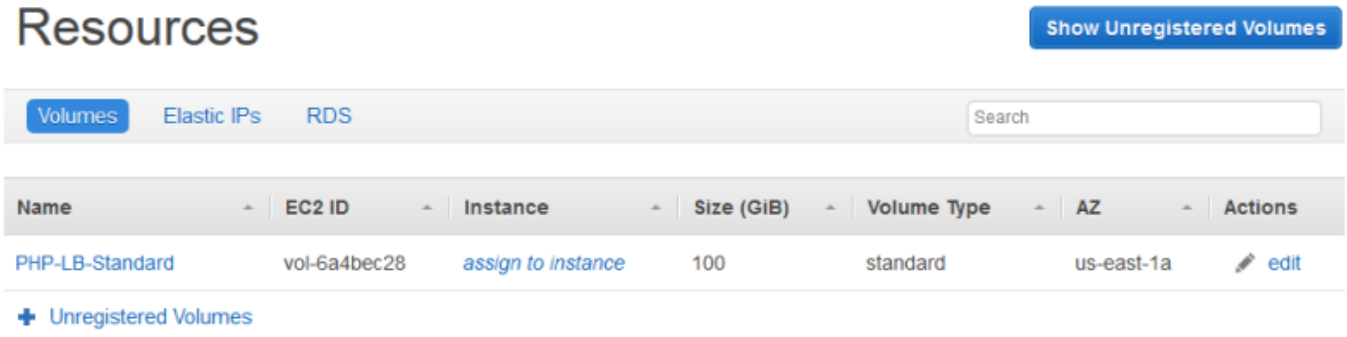
1. Buka tumpukan yang diinginkan dan klik Sumber daya di panel navigasi.
2. Klik Volume untuk menampilkan volume Amazon EBS yang tersedia. Awalnya, tumpukan tidak memiliki volume terdaftar, seperti yang ditunjukkan pada ilustrasi berikut.



3. Klik Tampilkan Volume yang Tidak Terdaftar untuk menampilkan volume Amazon EBS di akun Anda yang berada di wilayah tumpukan dan jika berlaku, VPC stack. Kolom Status menunjukkan apakah volume tersedia untuk digunakan. Jenis Volume menunjukkan apakah volume standar (`standard`), SSD Tujuan Umum (), PIOPS (`gp2io1`, diikuti oleh IOPS per nilai disk dalam tanda kurung), Throughput Optimized HDD (), atau Cold HDD (`st1`). `sc1`



4. Pilih volume yang sesuai dan klik Daftar ke Tumpukan. Halaman Sumber Daya sekarang mencantumkan volume yang baru terdaftar.



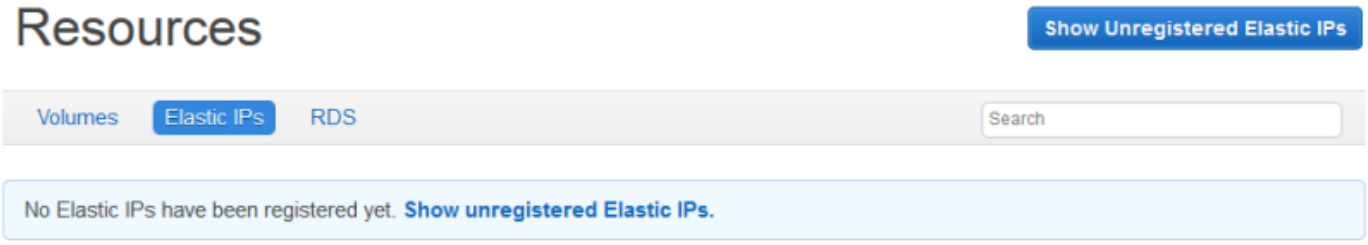
   Untuk mendaftarkan volume tambahan, klik Tampilkan Volume Tidak Terdaftar atau+ Volume Tidak Terdaftar dan ulangi prosedur ini.

Mendaftarkan Alamat IP Elastis dengan Stack

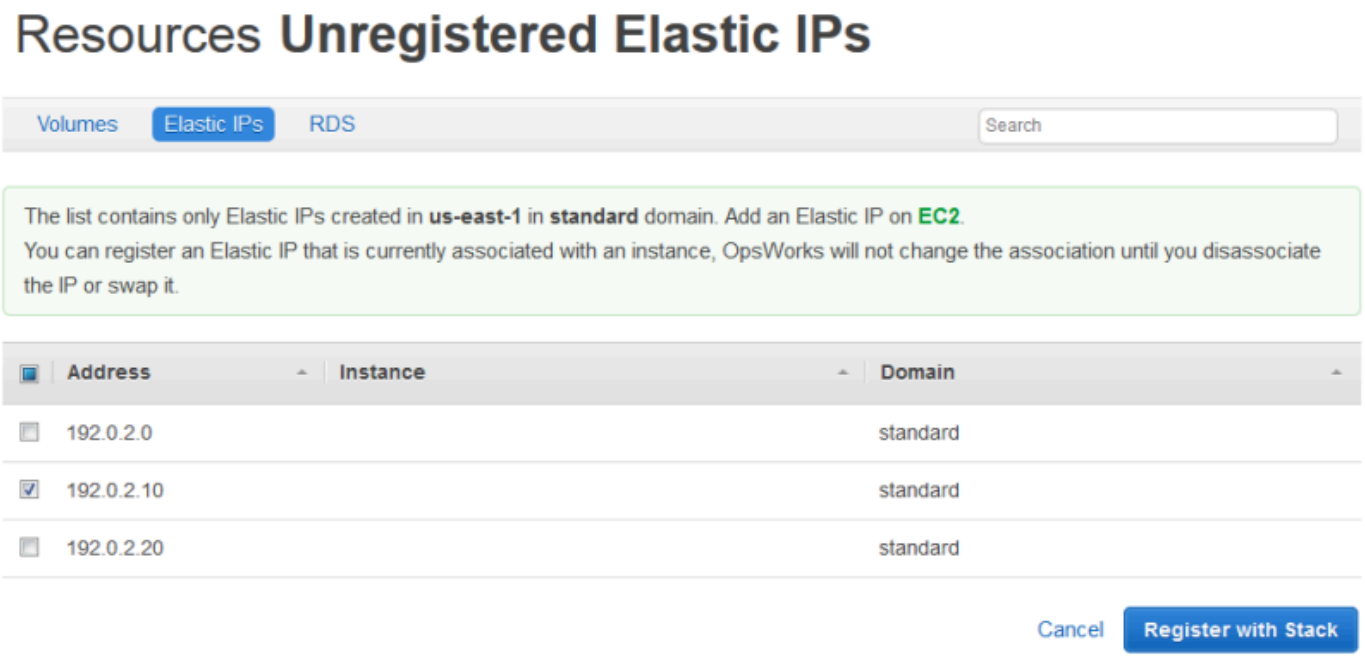
Gunakan prosedur berikut untuk mendaftarkan alamat IP Elastis.

Untuk mendaftarkan alamat IP Elastis

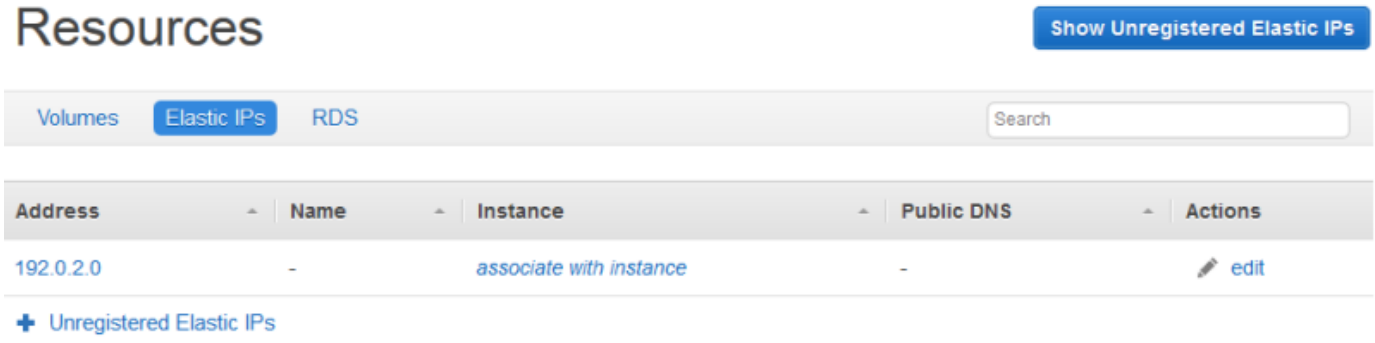
1. Buka halaman Sumber Daya tumpukan dan klik Elastic IPs untuk menampilkan alamat IP Elastis yang tersedia. Awalnya, tumpukan tidak memiliki alamat terdaftar, seperti yang ditunjukkan pada ilustrasi berikut.



2. Klik Tampilkan Elastis Tidak Terdaftar IPs untuk menampilkan alamat IP Elastis yang tersedia di akun Anda yang berada di wilayah tumpukan.



3. Pilih alamat yang sesuai dan klik Daftar ke Tumpukan. Ini mengembalikan Anda ke halaman Sumber Daya, yang sekarang mencantumkan alamat yang baru terdaftar.



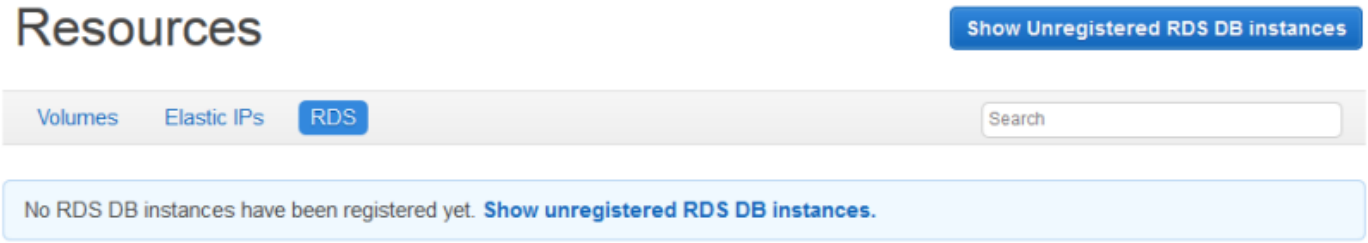
Untuk mendaftarkan alamat tambahan, klik Tampilkan Elastis Tidak Terdaftar IPs atau+Elastis Tidak Terdaftar IPs dan ulangi prosedur ini.

Mendaftarkan Instans Amazon RDS dengan Stack

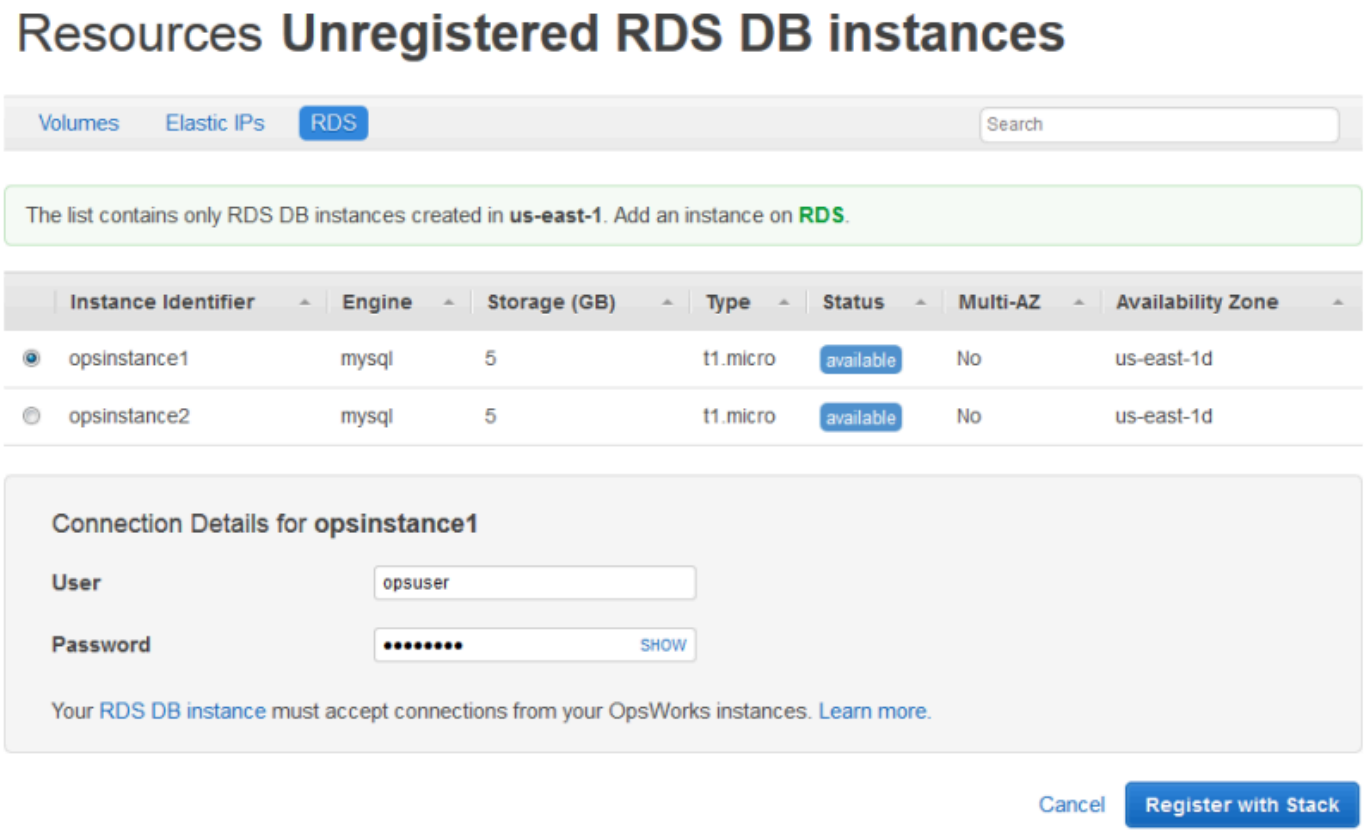
Gunakan prosedur berikut untuk mendaftarkan instans Amazon RDS.

Untuk mendaftarkan instans Amazon RDS

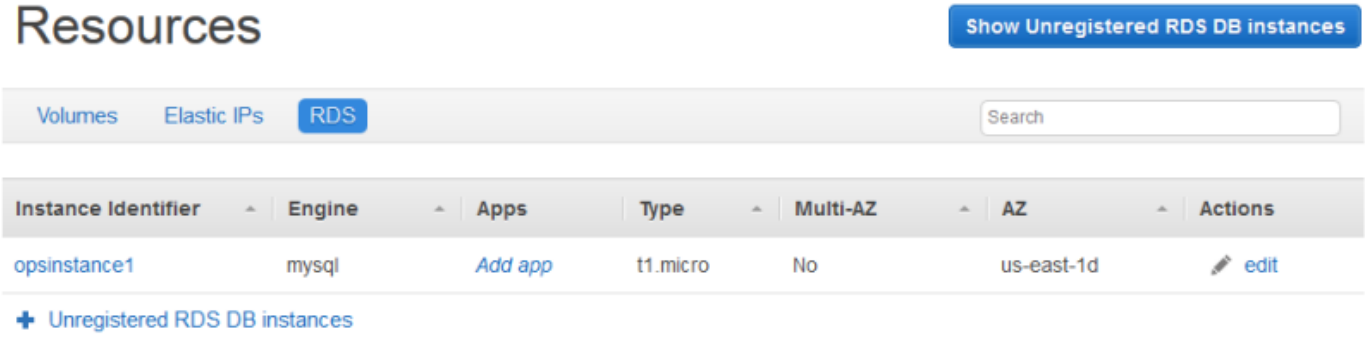
1. Buka halaman Sumber Daya tumpukan dan klik RDS untuk menampilkan instans Amazon RDS yang tersedia. Awalnya, tumpukan tidak memiliki instance terdaftar, seperti yang ditunjukkan pada ilustrasi berikut.



2. Klik Tampilkan instans RDS DB yang tidak terdaftar untuk menampilkan instans Amazon RDS yang tersedia di akun Anda yang berada di wilayah tumpukan.



3. Pilih contoh yang sesuai, masukkan nilai master user dan master password untuk User dan Password, dan klik Register to Stack. Ini mengembalikan Anda ke halaman Resources, yang sekarang mencantumkan instance yang baru terdaftar.



> **Important**
>
> Anda harus memastikan bahwa pengguna dan kata sandi yang Anda gunakan untuk mendaftarkan instans Amazon RDS sesuai dengan pengguna dan kata sandi yang valid. Jika tidak, aplikasi Anda tidak akan dapat terhubung ke instance.

Untuk mendaftarkan alamat tambahan, klik Tampilkan instans RDS DB yang tidak terdaftar atau +instans RDS DB yang tidak terdaftar dan ulangi prosedur ini. Untuk informasi selengkapnya tentang cara menggunakan instans Amazon RDS dengan AWS OpsWorks Stacks, lihat. Lapisan Layanan Amazon RDS

> **Note**
>
> Anda juga dapat mendaftarkan instans Amazon RDS melalui halaman Layers. Untuk informasi selengkapnya, lihat Lapisan Layanan Amazon RDS.

Melampirkan dan Memindahkan Sumber Daya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda mendaftarkan sumber daya dengan tumpukan, Anda dapat melampirkannya ke salah satu instance tumpukan. Anda juga dapat memindahkan sumber daya terlampir dari satu instance ke instance lainnya. Perhatikan hal berikut:

- Saat Anda melampirkan atau memindahkan volume Amazon EBS, instans yang terlibat dalam operasi harus offline. Jika instance yang Anda minati tidak ada di halaman Resources, buka halaman Instances dan hentikan instance. Setelah berhenti, Anda dapat kembali ke halaman Sumber Daya dan melampirkan atau memindahkan sumber daya.
- Saat Anda melampirkan atau memindahkan alamat IP Elastis, instans dapat online atau offline.
- Jika Anda menghapus instance, sumber daya terlampir tetap terdaftar dengan tumpukan. Anda kemudian dapat melampirkan sumber daya ke instance lain atau, jika Anda tidak lagi membutuhkannya, membatalkan pendaftaran sumber daya.

Topik

- Menetapkan Volume Amazon EBS ke Instance
- Mengaitkan Alamat IP Elastis dengan Instance
- Melampirkan Instans Amazon RDS ke Aplikasi

Menetapkan Volume Amazon EBS ke Instance

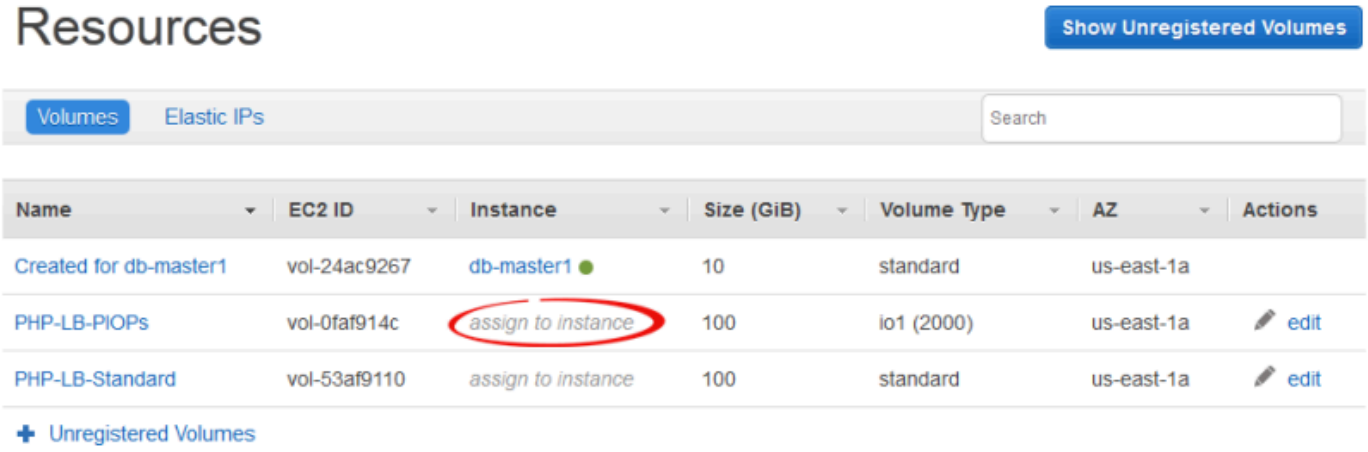
Note

Anda tidak dapat menetapkan volume Amazon EBS ke instans Windows.

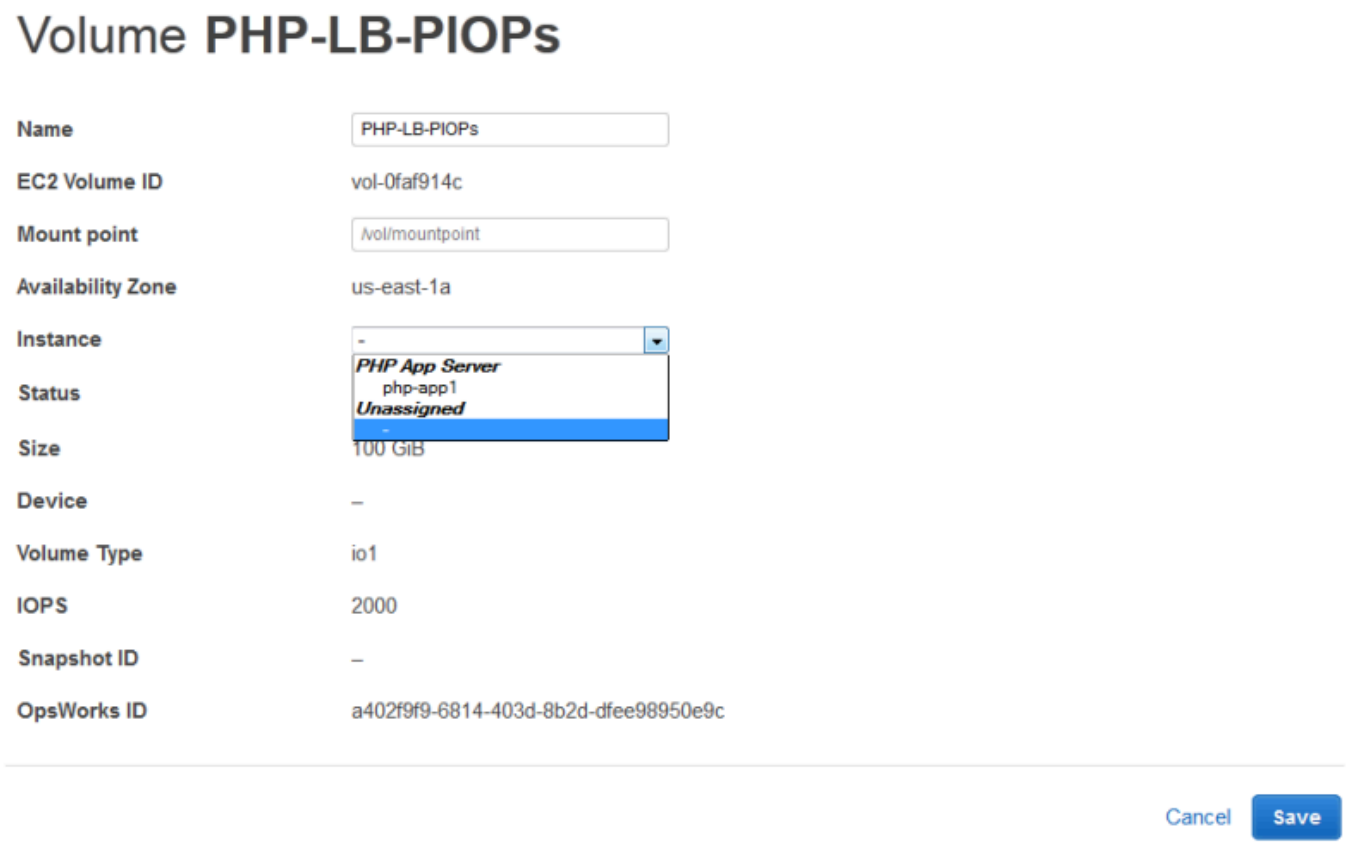
Anda dapat menetapkan volume Amazon EBS terdaftar ke instans dan memindahkannya dari satu instans ke instans lainnya, tetapi kedua instance tersebut harus offline.

Untuk menetapkan volume Amazon EBS ke instans

1. Pada halaman Resources, klik assign to instance di kolom Instance volume yang sesuai.



2. Pada halaman detail volume, pilih instance yang sesuai, tentukan nama volume dan titik pemasangan, dan klik Simpan untuk melampirkan volume ke instance.



> **Important**
>
> Jika Anda telah menetapkan volume penggunaan eksternal ke instans Anda, Anda harus menggunakan EC2 konsol Amazon, API, atau CLI untuk membatalkan penetapannya dari instance asli atau proses awal akan gagal.

Anda juga dapat menggunakan halaman detail untuk memindahkan volume Amazon EBS yang ditetapkan ke instans lain di tumpukan.

Untuk memindahkan volume Amazon EBS ke instans lain

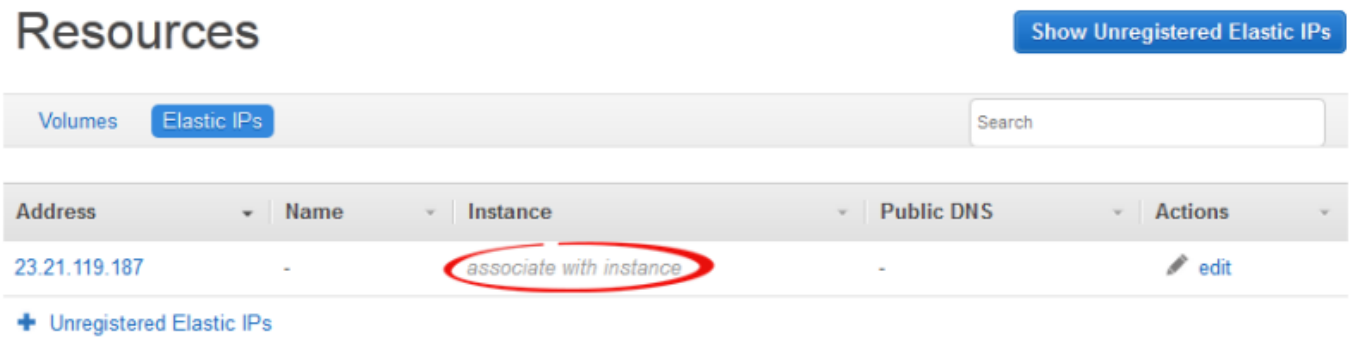
1. Pastikan kedua instance berada dalam keadaan offline.
2. Pada halaman Sumber daya, klik Volume dan kemudian klik edit di kolom Tindakan volume.
3. Lakukan salah satu hal berikut ini:
   - Untuk memindahkan volume ke instance lain di tumpukan, pilih instance yang sesuai dari daftar Instance dan klik Simpan.
   - Untuk memindahkan volume ke instance di tumpukan lain, deregister volume, daftarkan volume dengan tumpukan baru, dan lampirkan ke instance berita.

Mengaitkan Alamat IP Elastis dengan Instance

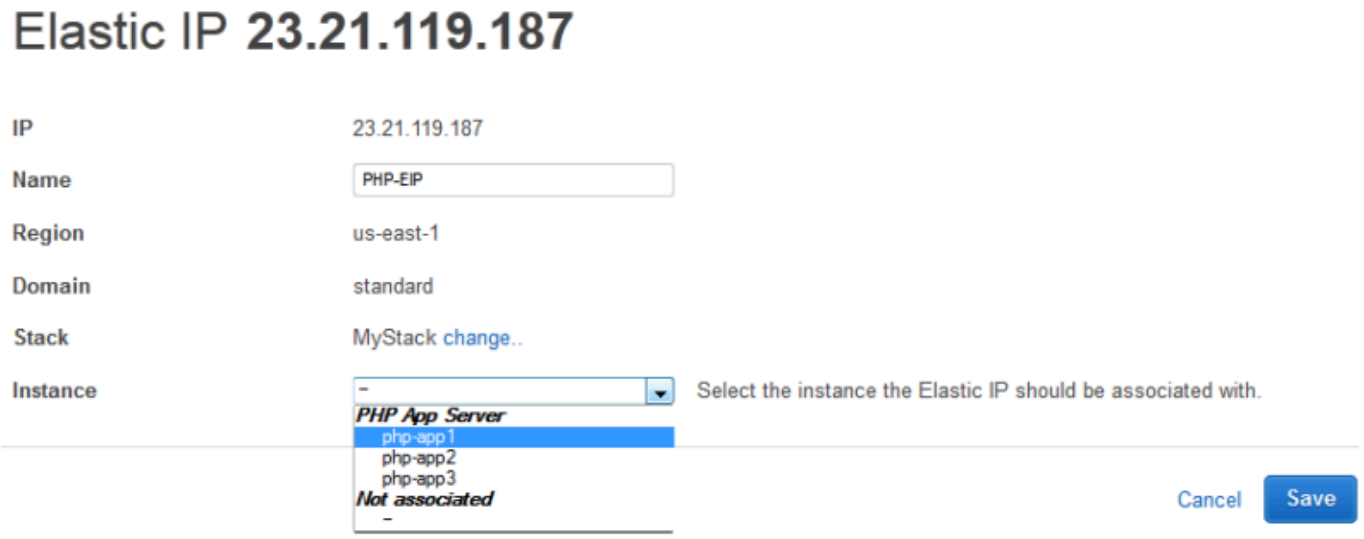
Anda dapat mengaitkan alamat IP Elastis terdaftar dengan instans dan memindahkannya dari satu instance ke instance lainnya, termasuk instance di tumpukan lain. Instans dapat online atau offline.

Untuk mengaitkan alamat IP Elastis dengan sebuah instans

1. Pada halaman Resources, klik kaitkan dengan instance di kolom Instance alamat yang sesuai.



2. Pada halaman detail alamat, pilih contoh yang sesuai, tentukan nama alamat, dan klik Simpan untuk mengaitkan alamat dengan instance.



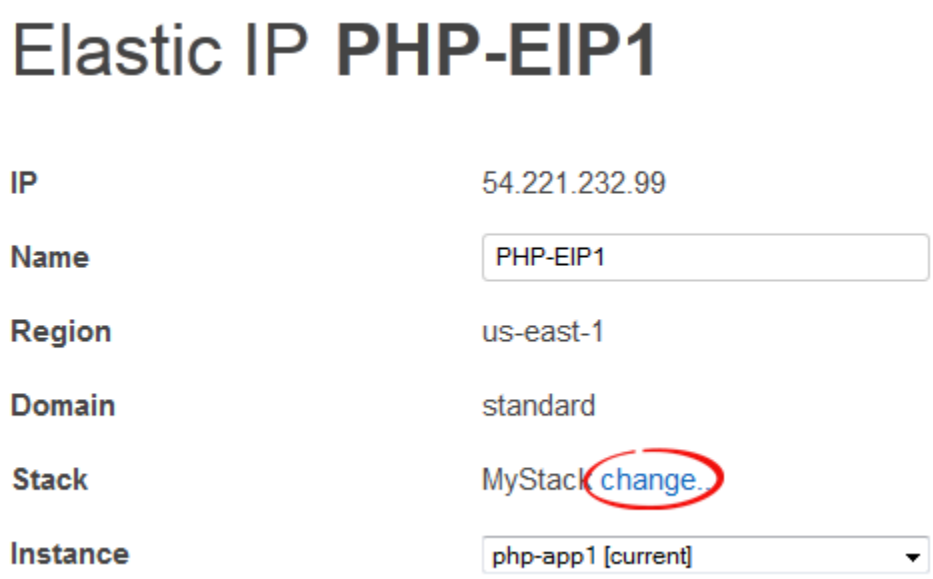
> **Note**
>
> Jika alamat IP Elastic saat ini dikaitkan dengan instans online lain, AWS OpsWorks Stacks secara otomatis menetapkan kembali alamat ke instance baru.

Anda juga dapat menggunakan halaman detail untuk memindahkan alamat IP Elastis terkait ke instance lain.

Untuk memindahkan alamat IP Elastis ke instance lain

1. Pada halaman Sumber Daya, klik Elastis IPs dan klik edit di kolom Tindakan alamat.
2. Lakukan salah satu hal berikut ini:

   - Untuk memindahkan alamat ke instance lain di tumpukan, pilih instance yang sesuai dari daftar Instance dan klik Simpan.
   - Untuk memindahkan alamat ke instance di tumpukan lain, klik ubah di pengaturan Stack untuk melihat daftar tumpukan yang tersedia. Pilih tumpukan dari daftar Stack dan instance dari daftar Instance. Kemudian klik Simpan.



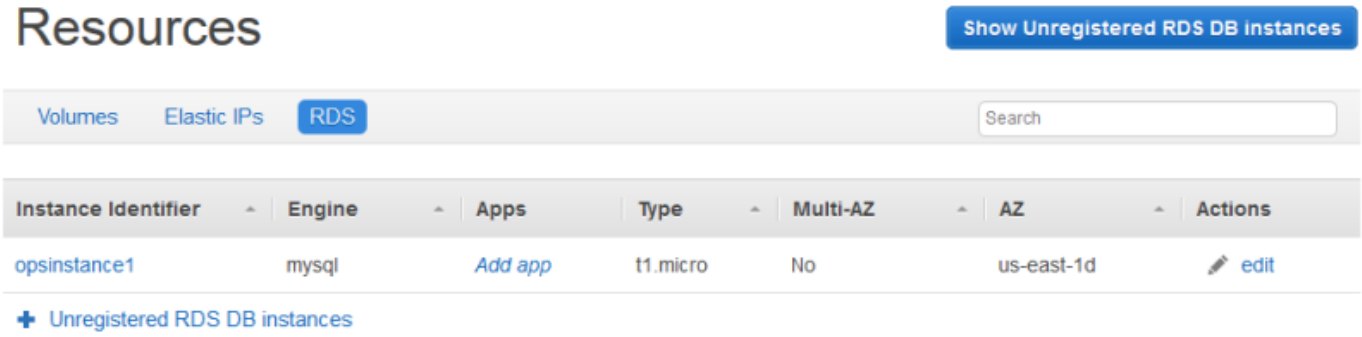
Setelah Anda melampirkan atau memindahkan alamat, AWS OpsWorks Stacks memicu peristiwa siklus hidup Konfigurasi untuk memberi tahu instance tumpukan tentang perubahan tersebut.

Melampirkan Instans Amazon RDS ke Aplikasi

Anda dapat melampirkan instans Amazon RDS ke satu atau beberapa aplikasi.

Untuk melampirkan instans Amazon RDS ke aplikasi

1. Pada halaman Sumber Daya, klik Tambahkan aplikasi di kolom Aplikasi instans yang sesuai.



2. Gunakan halaman Tambah Aplikasi untuk melampirkan instans Amazon RDS. Untuk informasi selengkapnya, lihat Menambahkan Aplikasi.

Karena Amazon RDS dapat dilampirkan ke beberapa aplikasi, tidak ada prosedur khusus untuk memindahkan instance dari satu aplikasi ke aplikasi lainnya. Cukup edit aplikasi pertama untuk

menghapus instance RDS atau edit aplikasi kedua untuk menambahkan instance RDS. Untuk informasi selengkapnya, lihat Mengedit Aplikasi.

Melepaskan Sumber Daya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Ketika Anda tidak lagi membutuhkan sumber daya terlampir, Anda dapat melepaskannya. Sumber daya ini tetap terdaftar dengan tumpukan dan dapat dilampirkan di tempat lain.

Topik

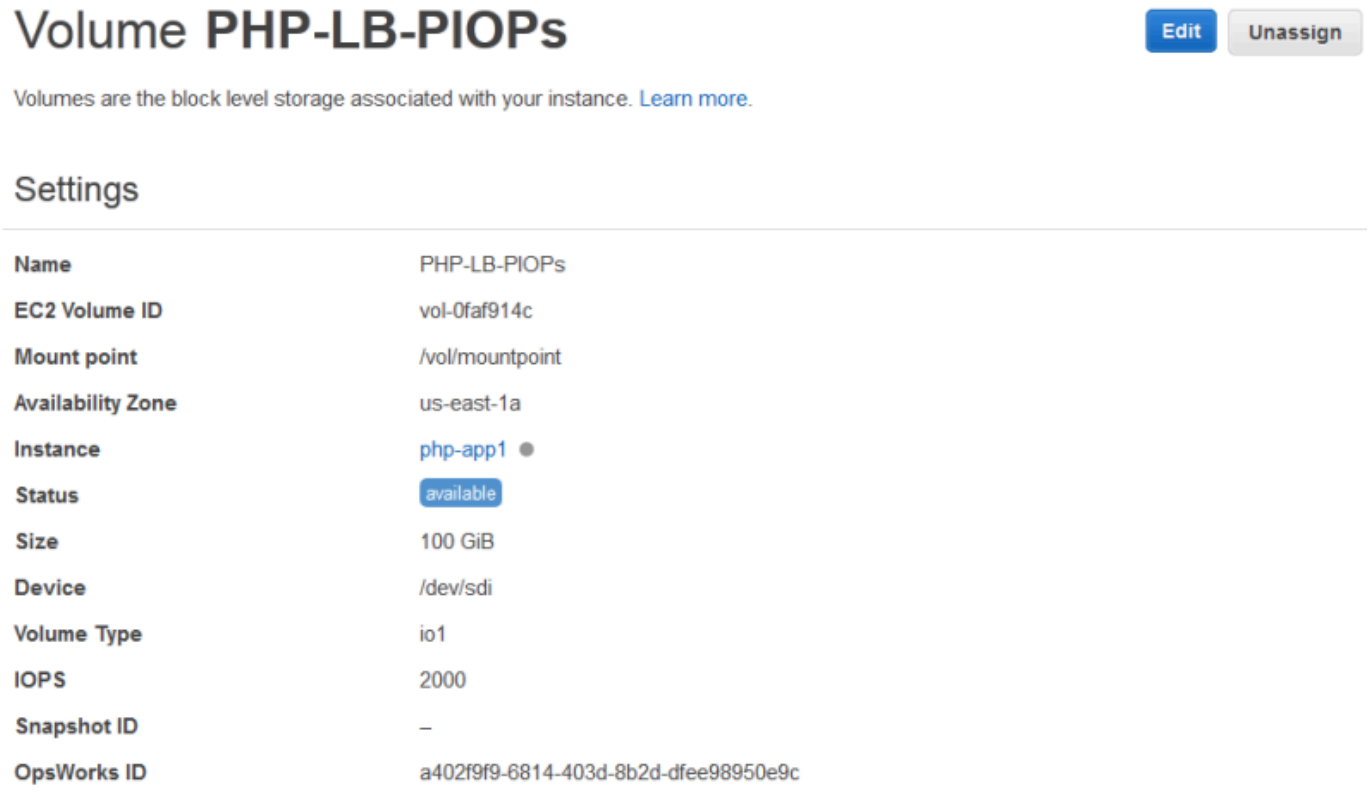
- Membatalkan penetapan Volume Amazon EBS
- Memutus Alamat IP Elastis
- Melepaskan Instans Amazon RDS

Membatalkan penetapan Volume Amazon EBS

Gunakan prosedur berikut untuk membatalkan penetapan volume Amazon EBS dari instancenya.

Untuk membatalkan penetapan volume Amazon EBS

1. Pastikan instans dalam keadaan offline.
2. Pada halaman Sumber Daya, klik Volume dan klik nama volume.
3. Pada halaman detail volume, klik Unassign.

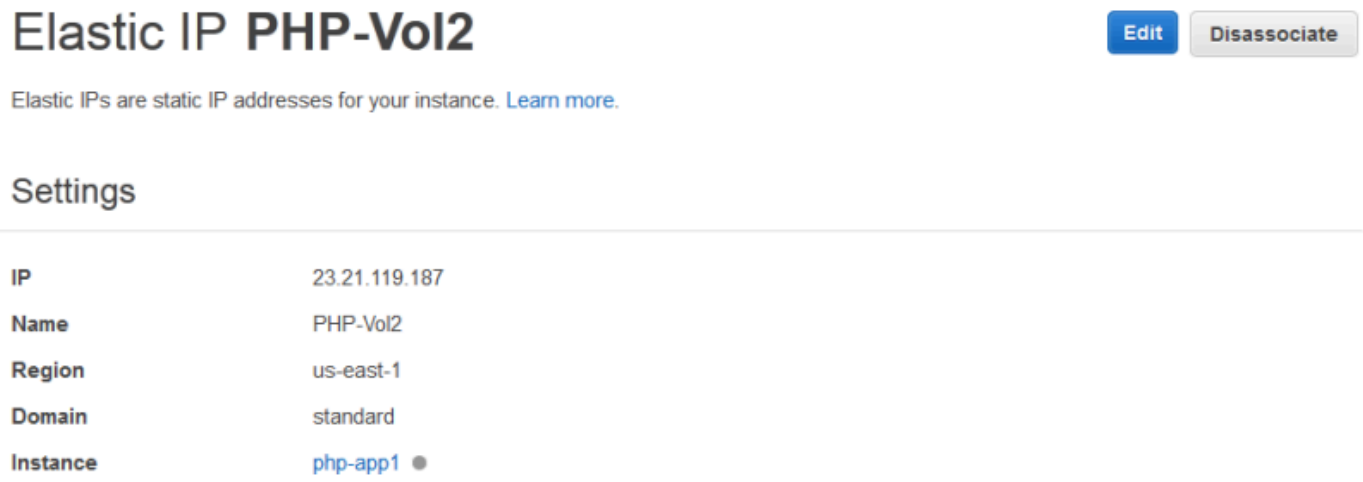


Memutus Alamat IP Elastis

Gunakan prosedur berikut untuk memisahkan alamat IP Elastis dari instance-nya.

Untuk memisahkan alamat IP Elastis

1. Pada halaman Sumber Daya, klik Elastis IPs dan klik edit di kolom Tindakan alamat.
2. Pada halaman detail alamat, klik Disassociate.



Setelah Anda memisahkan alamat, AWS OpsWorks Stacks memicu peristiwa siklus hidup Konfigurasi untuk memberi tahu instance tumpukan tentang perubahan tersebut.

Melepaskan Instans Amazon RDS

Gunakan prosedur berikut untuk melepaskan Amazon RDS dari aplikasi.

Untuk melepaskan instans Amazon RDS

1. Pada halaman Resources, klik RDS dan klik aplikasi yang sesuai di kolom Apps.
2. Klik Edit dan edit konfigurasi aplikasi untuk melepaskan instance. Untuk informasi selengkapnya, lihat Mengedit Aplikasi.

> **Note**
>
> Prosedur ini melepaskan Amazon RDS dari satu aplikasi. Jika instance dilampirkan ke beberapa aplikasi, Anda harus mengulangi prosedur ini untuk setiap aplikasi.

Sumber Daya Deregistering

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika Anda tidak perlu lagi memiliki sumber daya yang terdaftar dengan tumpukan, Anda dapat membatalkan pendaftarannya. Deregistrasi tidak menghapus sumber daya dari akun Anda; itu tetap ada dan dapat didaftarkan dengan tumpukan lain atau digunakan di luar Stacks. AWS OpsWorks Jika Anda ingin menghapus sumber daya sepenuhnya, Anda memiliki dua opsi:

- Jika IP Elastic atau sumber daya Amazon EBS dilampirkan ke instans, Anda dapat menghapus sumber daya saat menghapus instance.

Buka halaman Instans, klik hapus di kolom Tindakan instans, lalu pilih Hapus volume EBS instans atau Hapus IP Elastis instans.

- Batalkan pendaftaran sumber daya dan kemudian gunakan konsol Amazon atau EC2 Amazon RDS, API, atau CLI untuk menghapusnya.

Topik

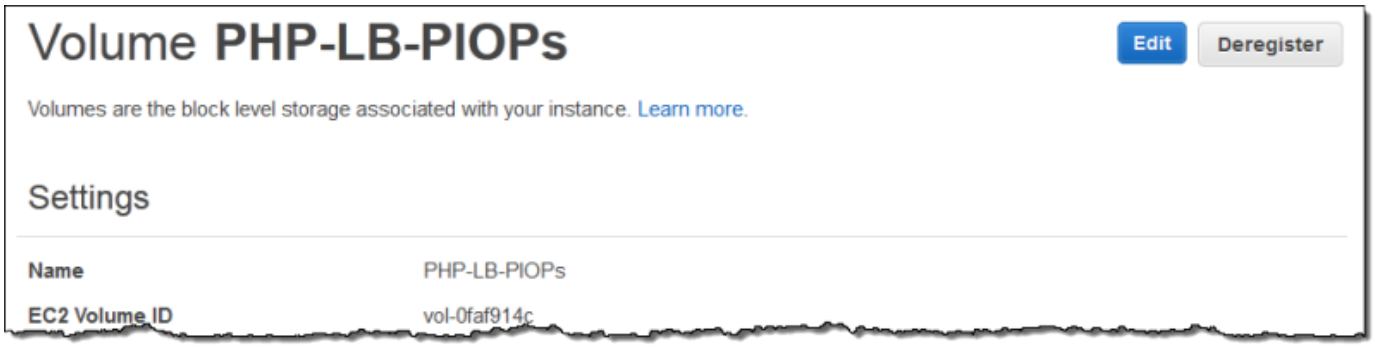
- Menderegistrasi Volume Amazon EBS
- Menderegistrasi Alamat IP Elastis
- Menderegistrasi Instans Amazon RDS

Menderegistrasi Volume Amazon EBS

Gunakan prosedur berikut untuk membatalkan pendaftaran volume Amazon EBS.

Untuk membatalkan pendaftaran volume Amazon EBS

1. Jika volume dilampirkan ke sebuah instance, batalkan penetapannya, seperti yang dijelaskan dalam. Membatalkan penetapan Volume Amazon EBS
2. Pada halaman Sumber Daya, klik nama volume di kolom Nama.
3. Pada halaman detail volume, klik Deregister.

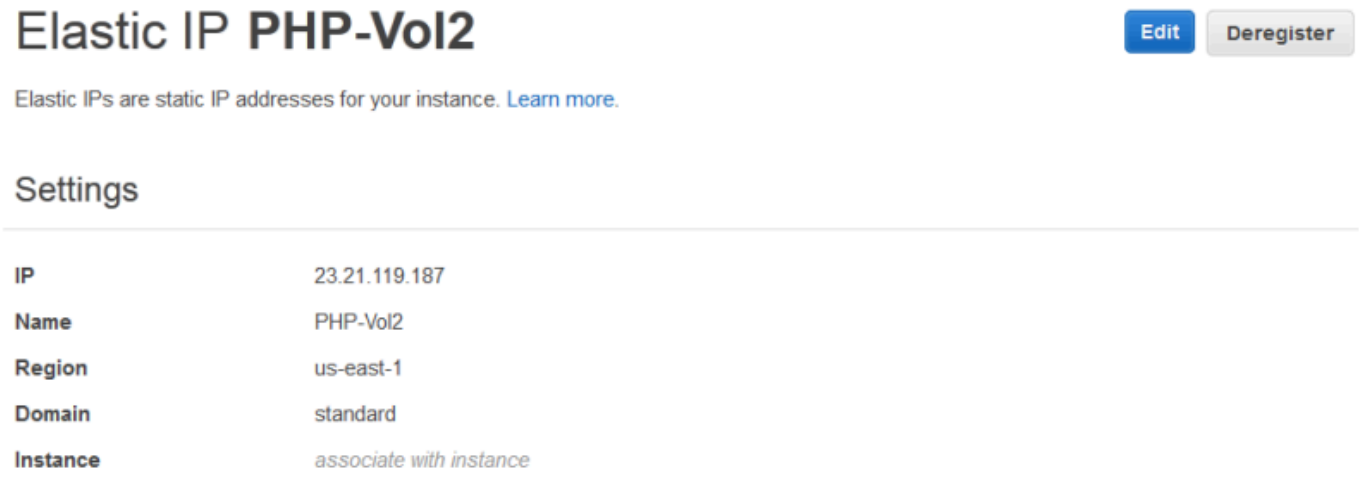


Menderegistrasi Alamat IP Elastis

Gunakan prosedur berikut untuk membatalkan pendaftaran alamat IP Elastis.

Untuk membatalkan pendaftaran alamat IP Elastis

1. Jika alamat dikaitkan dengan sebuah instance, pisahkan, seperti yang dijelaskan dalamMemutus Alamat IP Elastis.
2. Pada halaman Resources, klik Elastic IPs dan kemudian klik alamat IP di kolom Alamat.
3. Pada halaman detail alamat, klik Deregister.



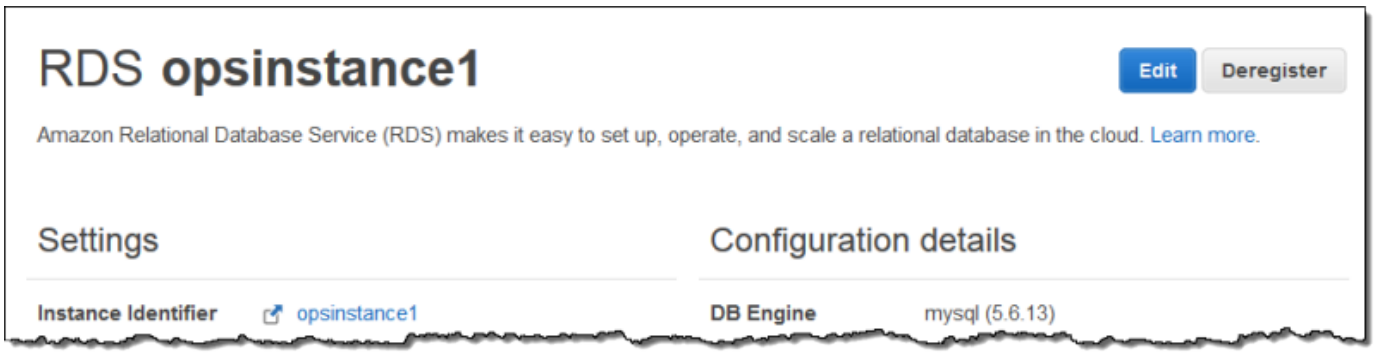
> **Note**
>
> Jika Anda hanya ingin mendaftarkan alamat IP Elastis dengan tumpukan yang berbeda, Anda harus membatalkan pendaftarannya dari tumpukan saat ini dan kemudian mendaftarkannya dengan tumpukan baru. Namun, Anda dapat memindahkan alamat IP Elastis terlampir ke instance di tumpukan lain secara langsung. Untuk informasi selengkapnya, lihat Melampirkan dan Memindahkan Sumber Daya.

Menderegistrasi Instans Amazon RDS

Gunakan prosedur berikut untuk membatalkan pendaftaran instans Amazon RDS.

Untuk membatalkan pendaftaran instans Amazon RDS

1. Jika instance dikaitkan dengan aplikasi, lepaskan, seperti yang dijelaskan dalamMelepaskan Sumber Daya.
2. Pada halaman Resources, klik RDS dan kemudian nama instance.
3. Pada halaman detail instance, klik Deregister.



Tanda

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tag dapat membantu Anda mengelompokkan sumber daya di tumpukan Chef 11.10, Chef 12, dan Chef 12.2, dan melacak biaya penggunaan sumber daya tersebut. AWS Manajemen Penagihan dan Biaya

Anda dapat menerapkan tag di tingkat tumpukan dan lapisan. Saat Anda membuat tag, Anda menerapkan tag ke setiap sumber daya dalam struktur yang ditandai. Misalnya, jika Anda menerapkan tag ke layer, Anda menerapkan tag ke setiap instance, volume Amazon EBS (kecuali root), atau penyeimbang beban Elastic Load Balancing di layer. Tag saat ini tidak dapat diterapkan ke root, atau default, volume EBS dari sebuah instance.

Tag adalah pasangan nilai kunci yang Anda tetapkan ke tumpukan atau lapisan di Stacks. AWS OpsWorks Setelah Anda membuat tag, buka konsol Billing and Cost Management untuk mengaktifkan tag yang ditentukan pengguna. Untuk informasi selengkapnya tentang cara mengaktifkan tag Anda dan menggunakannya untuk melacak dan mengelola biaya sumber daya AWS OpsWorks Stacks Anda, lihat Menggunakan Tag Alokasi Biaya dan Mengaktifkan Tag Alokasi Biaya yang Ditentukan Pengguna di Panduan Pengguna Billing and Cost Management.

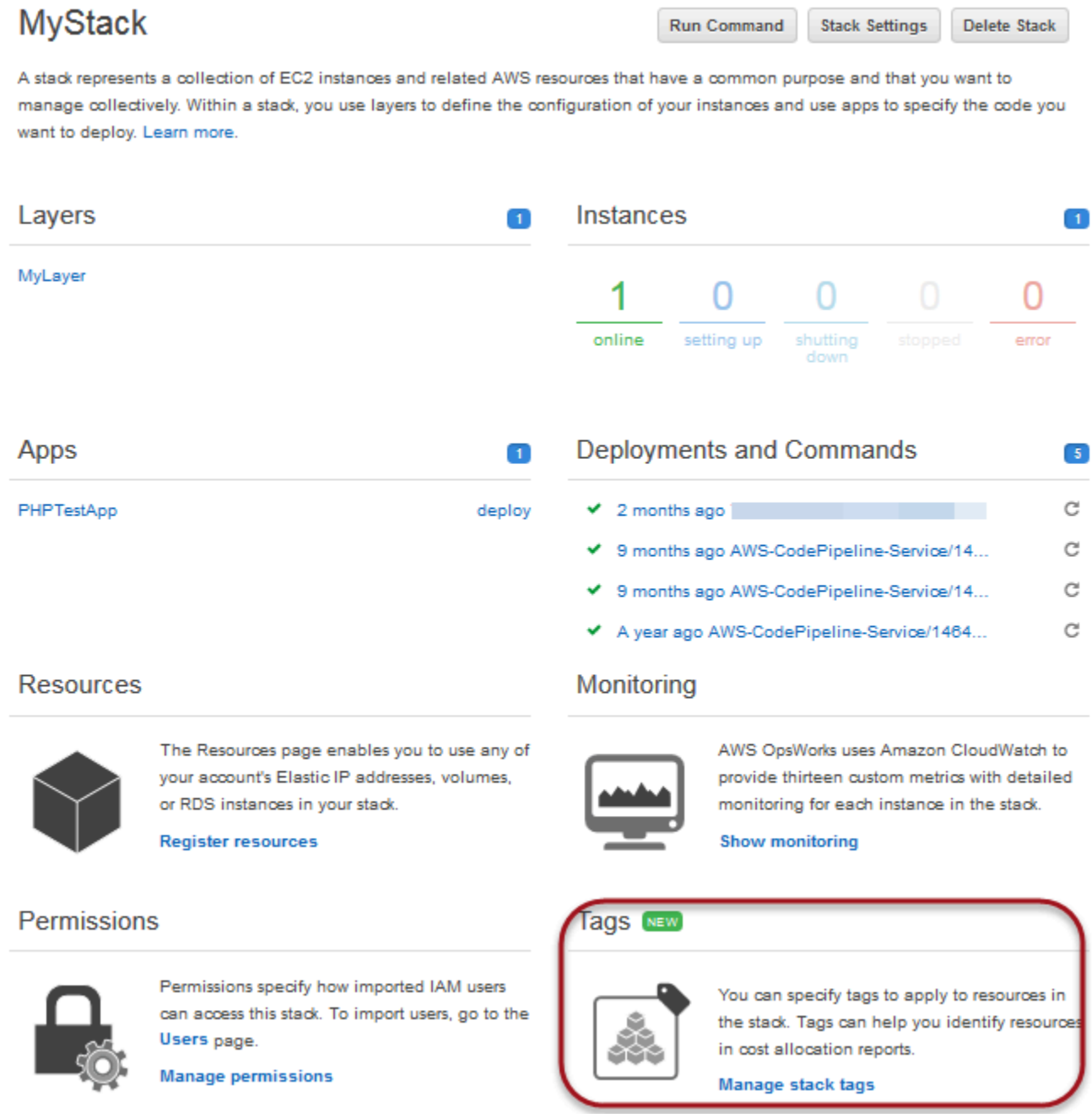
Tag bekerja dengan cara yang mirip dengan atribut khusus di AWS OpsWorks Stacks. Tag yang Anda terapkan ke tumpukan diwarisi oleh setiap lapisan di tumpukan. Pada tingkat lapisan, Anda dapat mengganti nilai (tetapi bukan nama kunci) dari tag yang diwarisi, dan menambahkan tag khusus lapisan baru. AWS OpsWorks menerapkan tag yang dihasilkan yang disetel ke semua sumber daya di lapisan. Saat Anda membuat sumber daya baru atau menetapkan sumber daya yang ada ke lapisan, sumber daya baru di lapisan ditandai dengan kumpulan tag yang sama.

Topik

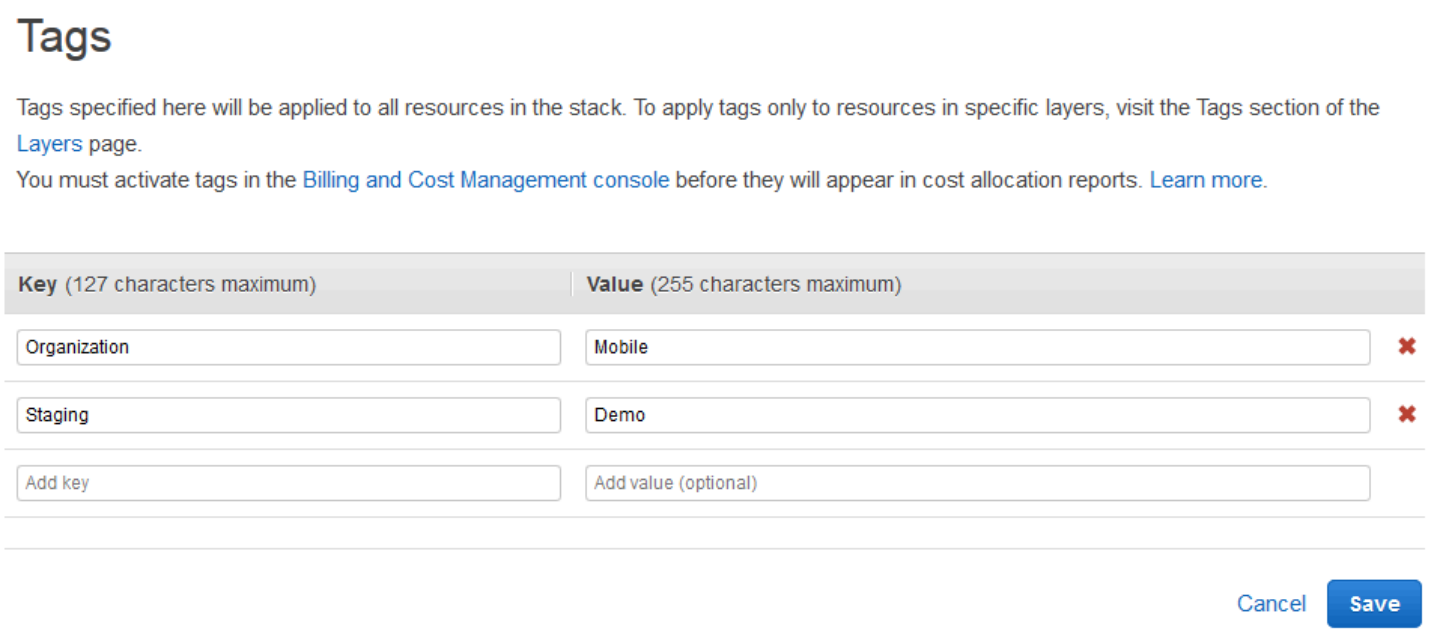
- Mengatur Tag di Stack Level
- Mengatur Tag di Tingkat Lapisan
- Mengelola Tag dengan AWS CLI
- Keterbatasan Tag

Mengatur Tag di Stack Level

Pada tingkat tumpukan, Anda dapat menambahkan dan mengelola tag dengan memilih Tag di halaman beranda tumpukan.

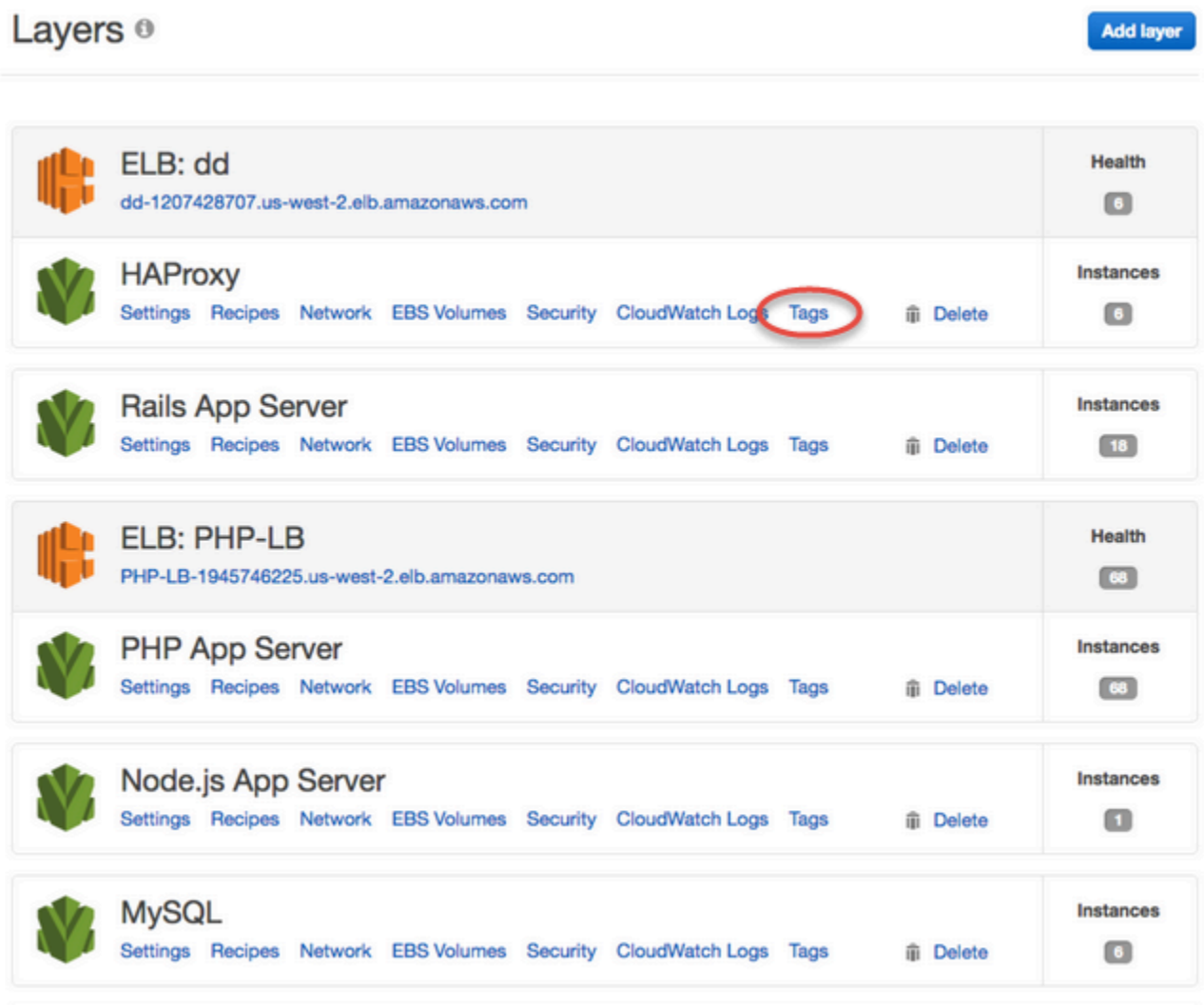


Pada halaman Tag, tambahkan tag sebagai pasangan nilai kunci. Screenshot berikut menunjukkan beberapa contoh tag. Anda dapat menghapus tag dengan memilih X merah di sebelah kanan pasangan kunci-nilai.

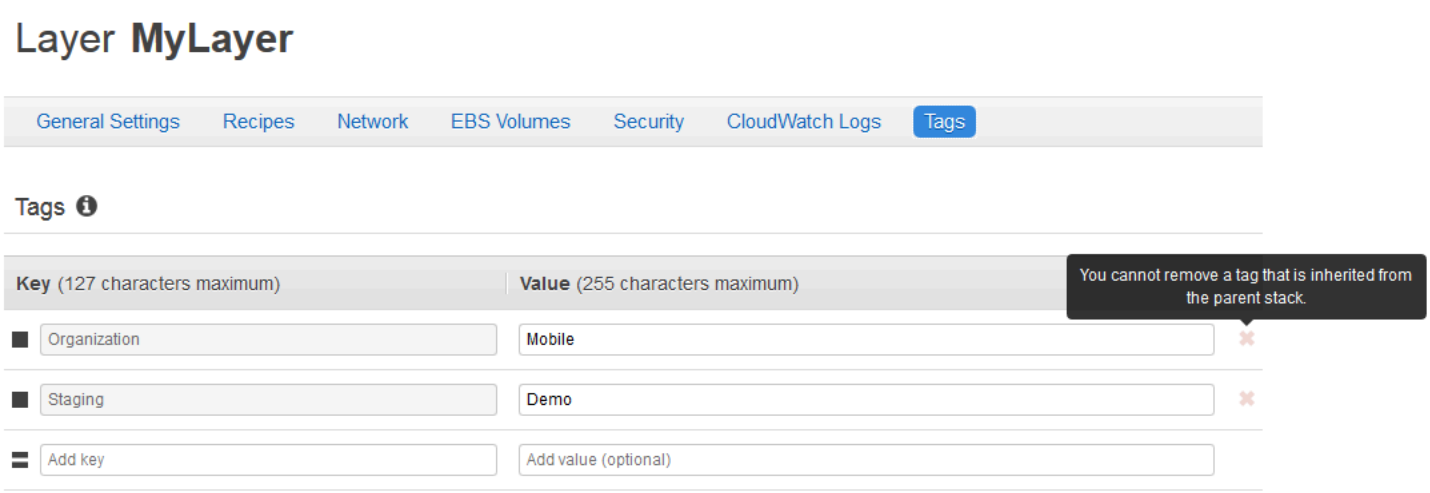


Mengatur Tag di Tingkat Lapisan

Pada tingkat lapisan, atur tag dengan memilih tab Tag. Anda dapat menemukan tab ini di halaman beranda Layers, dan halaman beranda untuk setiap lapisan individu.



Saat Anda mengubah atau menambahkan tag di tingkat lapisan, ketahuilah bahwa tag yang telah ditambahkan pada tingkat tumpukan induk diwarisi oleh lapisan dan sumber dayanya. Meskipun Anda dapat mengubah nilai tag yang diwariskan, Anda tidak dapat mengubah nama kunci, atau menghapus tag yang diwariskan. Ubah nama kunci atau hapus tag yang diwarisi dari tumpukan induk dalam pengaturan tumpukan. Tangkapan layar berikut menunjukkan contoh tag yang diwarisi dari tingkat tumpukan. Tag yang diwariskan berwarna abu-abu.



Untuk informasi selengkapnya tentang menambahkan tag ke tumpukan, lihatMembuat Tumpukan Baru. Untuk informasi selengkapnya tentang menambahkan tag ke lapisan, lihatMengedit Konfigurasi OpsWorks Layer.

Mengelola Tag dengan AWS CLI

Anda juga dapat menggunakan AWS CLI perintah untuk menambah dan menghapus tag di tingkat tumpukan dan lapisan. Untuk informasi selengkapnya tentang mengunduh dan menginstal AWS CLI, lihat Menginstal Antarmuka Baris AWS Perintah. Ingatlah untuk menambahkan `--region` parameter ke perintah Anda jika tumpukan yang ingin Anda tag tidak ada di wilayah default Anda. Layer saat ini ARNs tidak muncul di file AWS Management Console. Untuk mendapatkan ARN dari sebuah layer, jalankan perintah describe-layers.

Untuk menambahkan tag dengan menggunakan AWS CLI

- Pada prompt AWS CLI perintah, ketik perintah berikut, ganti ***stack_or_layer_ARN*** dan tentukan tag pasangan nilai kunci Anda, lalu tekan Enter. Tanda kutip ganda diloloskan dengan garis miring terbalik.

  ```
  aws opsworks tag-resource --resource-arn stack_or_layer_ARN --tags "{\"key\":
  \"value\",\"key\":\"value\"}"
  ```

  Berikut adalah contohnya.

  ```
  aws opsworks tag-resource --resource-arn arn:aws:opsworks:us-
  east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tags "{\"Stage\":
  \"Production\",\"Organization\":\"Mobile\"}"
  ```

Untuk menghapus tag dengan menggunakan AWS CLI

- Pada prompt AWS CLI perintah, ketik berikut ini, lalu tekan Enter.

```
aws opsworks untag-resource --resource-arn stack_or_layer_ARN --tag-keys "[\"key\",
\"key\"]"
```

Untuk menghapus tag, Anda hanya menentukan kunci tag yang ingin Anda hapus. Berikut adalah contohnya.

```
aws opsworks untag-resource --resource-arn arn:aws:opsworks:us-
east-2:800000000003:stack/500b99c0-ec00-4cgg-8a0d-1000000jjd1b --tag-keys "[\"Stage
\",\"Organization\"]"
```

> **Note**
>
> Anda tidak dapat menghapus tag yang diwariskan (tag yang ditambahkan pada tingkat tumpukan induk) dari lapisan. Hapus tag yang diwariskan dari tumpukan sebagai gantinya.

Keterbatasan Tag

Ingatlah batasan berikut saat Anda membuat tag.

- AWS OpsWorks Tumpukan membatasi jumlah tag yang ditentukan pengguna pada tingkat tumpukan dan lapisan hingga 40, termasuk tag yang ditentukan pengguna yang diwarisi dari tingkat induk. Ini menyisakan 10 slot yang tersedia untuk tag default yang ditambahkan dengan`opsworks:`, dan tag yang diatur oleh proses lain AWS . Maksimal 50 tag diperbolehkan pada sumber daya, termasuk tag yang ditentukan pengguna dan default yang dibuat oleh. AWS
- Tombol tag tidak dapat dimulai dengan**`aws:`**, **`opsworks:`** atau**`rds:`**. Jangan gunakan **`name`** atau **`Name`** sebagai kunci tag, karena **`Name`** dicadangkan oleh AWS OpsWorks Stacks.
- Sebuah kunci dapat maksimal 127 karakter, dan hanya dapat berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: `+ - = . _ : /`
- Nilai dapat berupa maksimum 255 karakter, dan hanya berisi huruf Unicode, angka, atau pemisah, atau karakter khusus berikut: `+ - = . _ : /`

Pemantauan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat memantau tumpukan Anda dengan cara berikut.

- AWS OpsWorks Stacks menggunakan Amazon CloudWatch untuk menyediakan tiga belas metrik kustom dengan pemantauan terperinci untuk setiap instance dalam tumpukan.
- AWS OpsWorks Stacks terintegrasi dengan AWS CloudTrail log setiap panggilan AWS OpsWorks Stacks API dan menyimpan data dalam bucket Amazon S3.
- Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau sistem tumpukan, aplikasi, dan log kustom Anda.

Topik

- Memantau Tumpukan menggunakan Amazon CloudWatch
- Logging AWS OpsWorks Stacks API Panggilan dengan AWS CloudTrail
- Menggunakan CloudWatch Log Amazon dengan AWS OpsWorks Tumpukan
- Memantau Tumpukan menggunakan Amazon CloudWatch Events

Memantau Tumpukan menggunakan Amazon CloudWatch

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks menggunakan Amazon CloudWatch (CloudWatch) untuk menyediakan pemantauan untuk tumpukan.

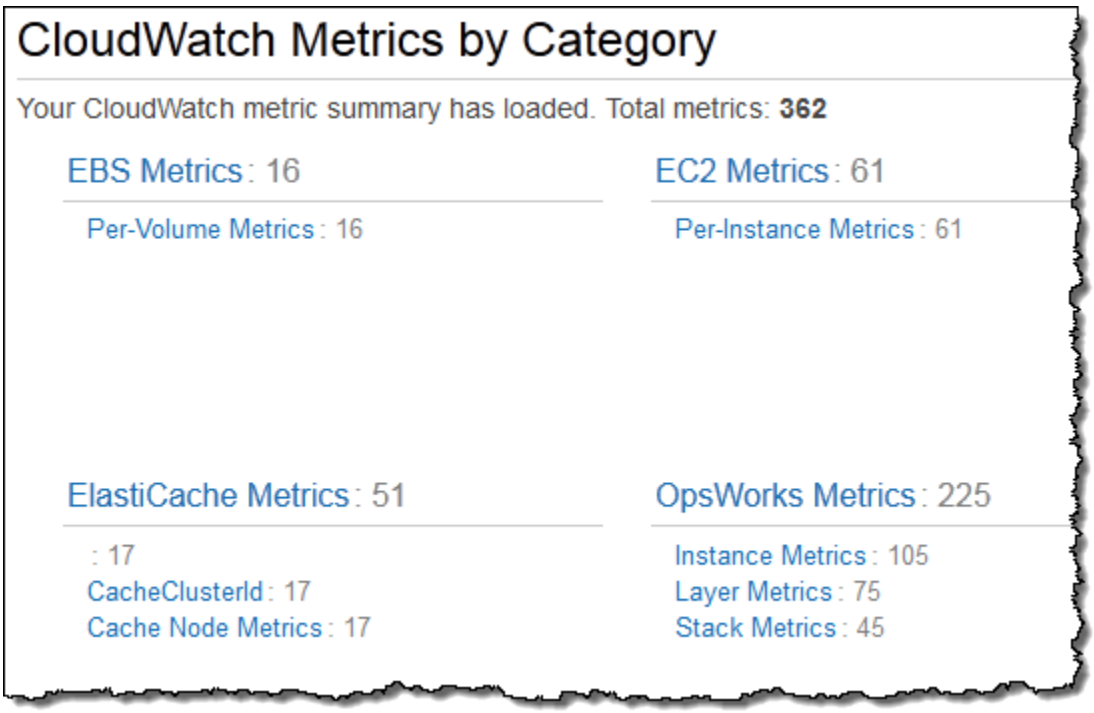
- Untuk tumpukan Linux, AWS OpsWorks Stacks mendukung tiga belas metrik kustom untuk memberikan pemantauan terperinci untuk setiap instance dalam tumpukan dan merangkum data untuk kenyamanan Anda di halaman Monitoring.
- Untuk tumpukan Windows, Anda dapat memantau EC2 metrik Amazon standar untuk instans Anda dengan konsol. CloudWatch

  Halaman Monitoring tidak menampilkan metrik Windows.

Halaman Monitoring menampilkan metrik untuk seluruh tumpukan, lapisan, atau instance. AWS OpsWorks Metrik tumpukan berbeda dari metrik Amazon EC2 . Anda juga dapat mengaktifkan metrik tambahan melalui CloudWatch konsol, tetapi biasanya memerlukan biaya tambahan. Anda juga dapat melihat data yang mendasarinya di CloudWatch konsol, sebagai berikut:

Untuk melihat metrik OpsWorks kustom di CloudWatch

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada bilah navigasi, pilih wilayah tumpukan.
3. Di panel navigasi, pilih Metrik.
4. Di OpsWorks Metrik, pilih Metrik Instance, Metrik Lapisan, atau Metrik Tumpukan.



> **Note**
>
> AWS OpsWorks Stacks mengumpulkan metrik dengan menjalankan proses pada setiap instance (agen instance). Karena CloudWatch mengumpulkan metrik secara berbeda, menggunakan hypervisor, nilai di CloudWatch konsol mungkin sedikit berbeda dari nilai yang sesuai pada halaman Pemantauan di konsol Stacks. AWS OpsWorks

Anda juga dapat menggunakan CloudWatch konsol untuk mengatur alarm. Untuk informasi selengkapnya tentang cara membuat alarm, lihat Membuat CloudWatch Alarm Amazon. Untuk daftar metrik CloudWatch kustom, lihat AWS OpsWorks Metrics and Dimensions. Untuk informasi selengkapnya, lihat Amazon CloudWatch.

Topik

- AWS OpsWorks Tumpukan Metrik
- Dimensi untuk AWS OpsWorks Metrik Tumpukan
- Metrik Stack
- Metrik Lapisan
- Metrik Instance

AWS OpsWorks Tumpukan Metrik

AWS OpsWorks Stacks mengirimkan metrik berikut ke CloudWatch setiap lima menit.

Metrik CPU

Metrik	Deskripsi
`cpu_idle`	Persentase waktu saat CPU tidak digunakan. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`cpu_nice`	Persentase waktu CPU menangani proses dengan `nice` nilai positif, yang memiliki prioritas penjadwalan yang lebih rendah. Untuk informasi lebih lanjut tentang ukuran ini, lihat bagus (Unix). Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`cpu_steal`	Karena AWS mengalokasikan sumber daya CPU hypervisor di antara peningkatan jumlah instance, beban virtualisasi meningkat, dan dapat memengaruhi seberapa sering hypervisor dapat melakukan pekerjaan yang diminta pada sebuah instance.

Metrik	Deskripsi
	`cpu_steal` mengukur persentase waktu instance menunggu hypervisor mengalokasikan sumber daya CPU fisik. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`cpu_system`	Persentase waktu CPU menangani operasi sistem. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`cpu_user`	Persentase waktu CPU menangani operasi pengguna. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada

Metrik	Deskripsi
`cpu_waitio`	Persentase waktu CPU menunggu operasi input/output. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada

Metrik Memori

Metrik	Deskripsi
`memory_buffers`	Jumlah memori buffer. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`memory_cached`	Jumlah memori yang di-cache. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId.

Metrik	Deskripsi
	Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`memory_free`	Jumlah memori bebas. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`memory_swap`	Jumlah ruang swap. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada

Metrik	Deskripsi
`memory_total`	Total jumlah memori. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`memory_used`	Jumlah memori yang digunakan. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada

Muatkan Metrik

Metrik	Deskripsi
`load_1`	Beban rata-rata selama satu menit jendela. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`.

Metrik	Deskripsi
	Satuan: Tidak ada
`load_5`	Beban rata-rata selama jendela lima menit. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada
`load_15`	Beban rata-rata lebih dari jendela 15 menit. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId. Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada

Metrik Proses

Metrik	Deskripsi
`procs`	Jumlah proses aktif. Dimensi yang IDs Valid: Sumber daya individu yang Anda lihat metriknya: StackId, LayerId, atau InstanceId.

Metrik	Deskripsi
	Statistik yang Valid: `Average`,`Minimum`,`Maximum`,`Sum`,, atau`Data Samples`. Satuan: Tidak ada

Dimensi untuk AWS OpsWorks Metrik Tumpukan

AWS OpsWorks Metrik tumpukan menggunakan namespace AWS OpsWorks Stacks, dan menyediakan metrik untuk dimensi berikut:

Dimensi	Deskripsi
`StackId`	Nilai rata-rata untuk tumpukan.
`LayerId`	Nilai rata-rata untuk lapisan.
`InstanceId`	Nilai rata-rata untuk sebuah instance.

Metrik Stack

Untuk melihat ringkasan metrik untuk seluruh tumpukan, pilih tumpukan di Dasbor AWS OpsWorks Tumpukan dan kemudian klik Pemantauan di panel navigasi. Contoh berikut adalah untuk tumpukan dengan PHP dan lapisan DB.



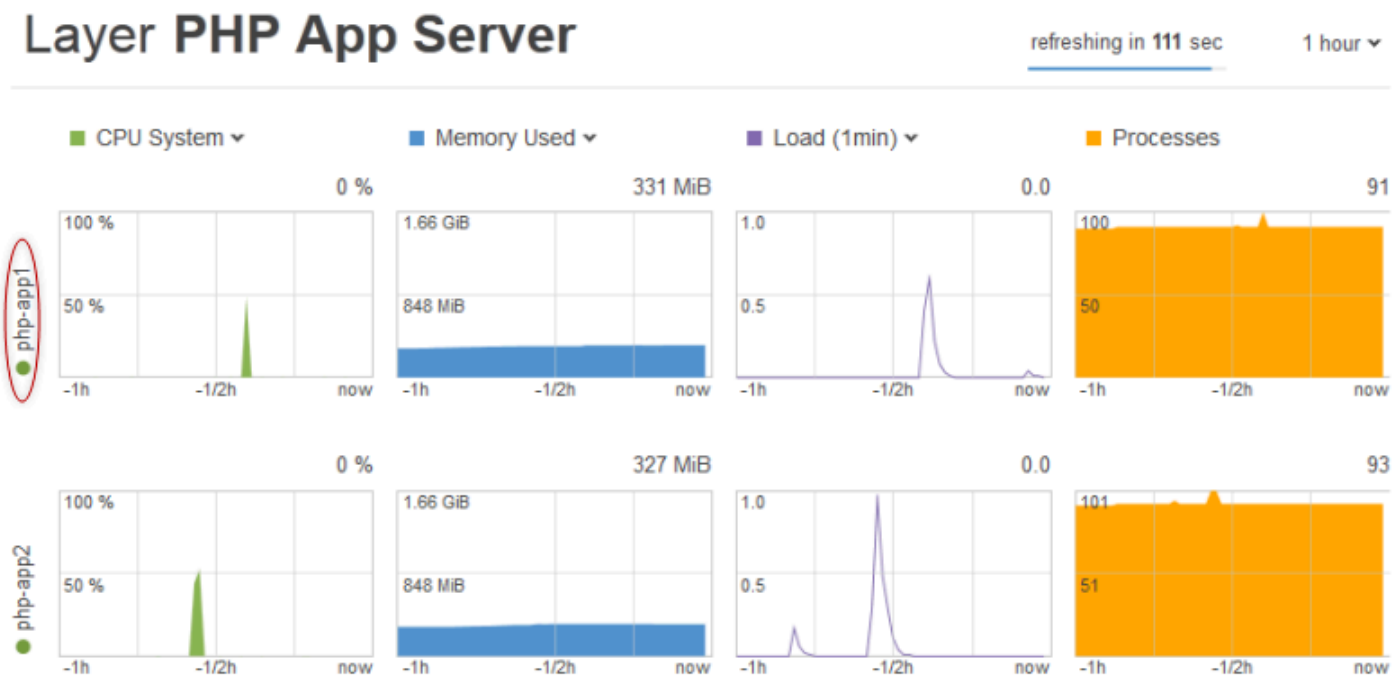
Tampilan tumpukan menampilkan grafik dari empat jenis metrik untuk setiap lapisan selama periode waktu tertentu: 1 jam, 8 jam, 24 jam, 1 minggu, atau 2 minggu. Perhatikan hal berikut:

- AWS OpsWorks Tumpukan memperbarui grafik secara berkala; penghitung waktu mundur di kanan atas menunjukkan waktu yang tersisa hingga pembaruan berikutnya,
- Jika sebuah layer memiliki lebih dari satu instance, grafik menampilkan nilai rata-rata untuk layer tersebut.
- Anda dapat menentukan periode waktu dengan mengklik daftar di kanan atas dan memilih nilai pilihan Anda.

Untuk setiap jenis metrik, Anda dapat menggunakan daftar di bagian atas grafik untuk memilih metrik tertentu yang ingin Anda lihat.

Metrik Lapisan

Untuk melihat metrik untuk layer tertentu, klik nama layer dalam tampilan Monitoring Layers. Contoh berikut menunjukkan metrik untuk lapisan PHP, yang memiliki dua contoh.

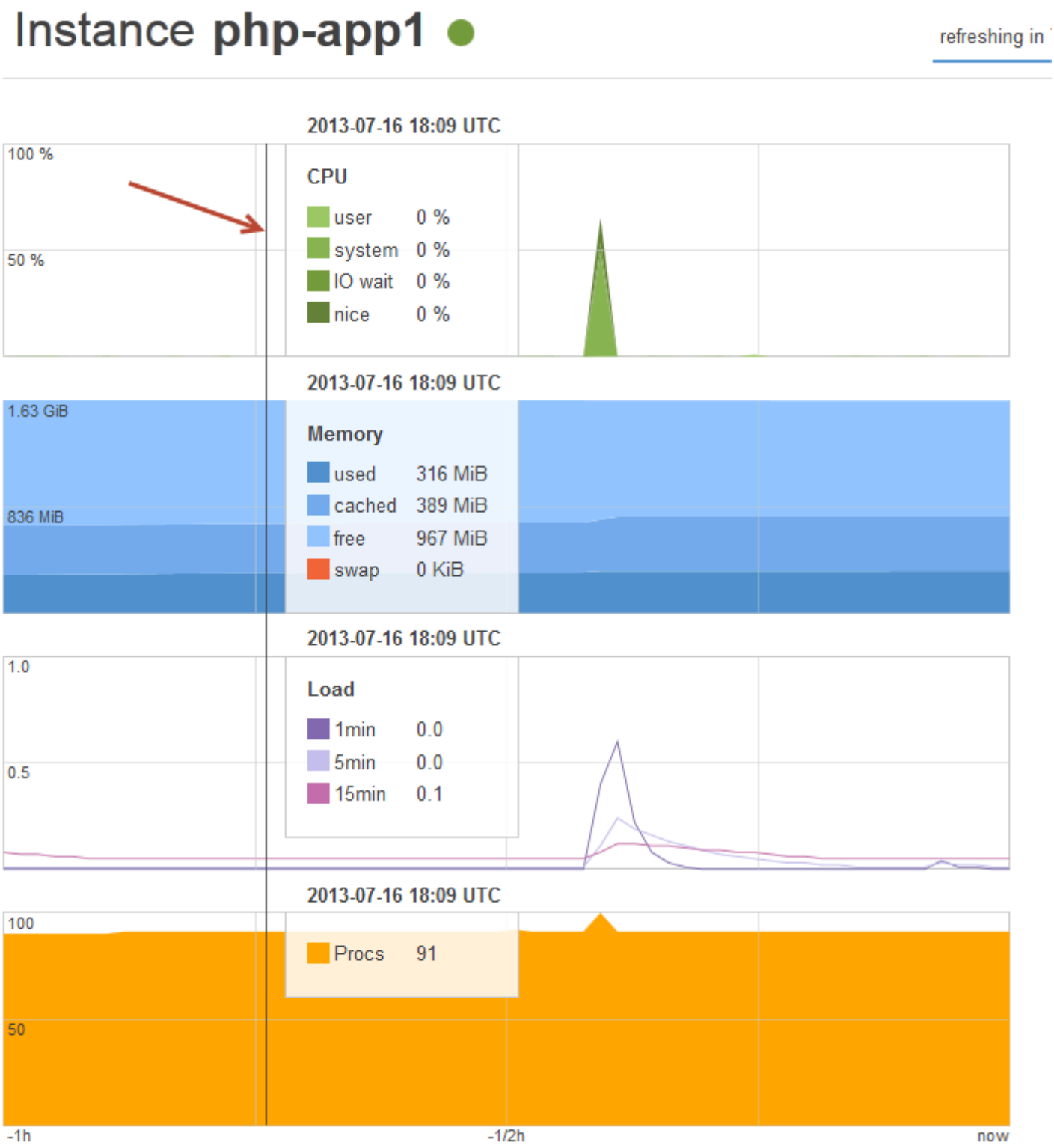


Jenis metrik sama dengan metrik tumpukan, dan untuk setiap jenis, Anda dapat menggunakan daftar di bagian atas grafik untuk memilih metrik tertentu yang ingin Anda lihat.

> **Note**
>
> Anda juga dapat menampilkan metrik lapisan dengan membuka halaman detail layer dan mengklik Monitoring di kanan atas.

Metrik Instance

Untuk melihat metrik untuk instance tertentu, klik nama instance di tampilan pemantauan lapisan. Contoh berikut menunjukkan metrik untuk instance php-app1 lapisan 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

Grafik merangkum semua metrik yang tersedia untuk setiap jenis metrik. Untuk mendapatkan nilai yang tepat untuk titik waktu tertentu, gunakan mouse Anda untuk memindahkan slider (ditunjukkan oleh panah merah pada ilustrasi sebelumnya) ke posisi yang sesuai.

> **Note**
>
> Anda juga dapat menampilkan metrik instance dengan membuka halaman detail instans dan memilih Monitoring di kanan atas.

Logging AWS OpsWorks Stacks API Panggilan dengan AWS CloudTrail

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh identitas IAM, atau AWS layanan di AWS OpsWorks Stacks. CloudTrail menangkap semua panggilan API untuk AWS OpsWorks Stacks sebagai peristiwa, termasuk panggilan dari konsol AWS OpsWorks Stacks dan dari panggilan kode ke Stacks. AWS OpsWorks APIs Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk AWS OpsWorks Stacks. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk AWS OpsWorks Stacks, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

AWS OpsWorks Informasi Tumpukan di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di AWS OpsWorks Tumpukan, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Peristiwa. Anda dapat melihat, mencari, dan

mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk AWS OpsWorks Stacks, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, ketika Anda membuat jejak di konsol, jejak ini diterapkan ke semua Wilayah. Trail mencatat peristiwa dari semua wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

Semua tindakan AWS OpsWorks Stacks dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API AWS OpsWorks Stacks. Misalnya, panggilan ke`CreateLayer`,`DescribeInstances`, dan `StartInstance` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan tersebut dibuat dengan kredensial root atau pengguna IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

Memahami AWS OpsWorks Entri Berkas Log Stacks

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber mana pun dan mencakup informasi tentang tindakan yang diminta,

tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateLayer` tindakan.


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


Menggunakan CloudWatch Log Amazon dengan AWS OpsWorks Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk menyederhanakan proses pemantauan log pada beberapa instance, AWS OpsWorks Stacks mendukung Amazon Logs. CloudWatch Anda mengaktifkan CloudWatch Log pada tingkat lapisan di AWS OpsWorks Stacks. CloudWatch Integrasi log berfungsi dengan tumpukan berbasis Linux Chef 11.10 dan Chef 12. Anda dikenakan biaya tambahan saat mengaktifkan CloudWatch Log, jadi tinjau CloudWatchHarga Amazon sebelum memulai.

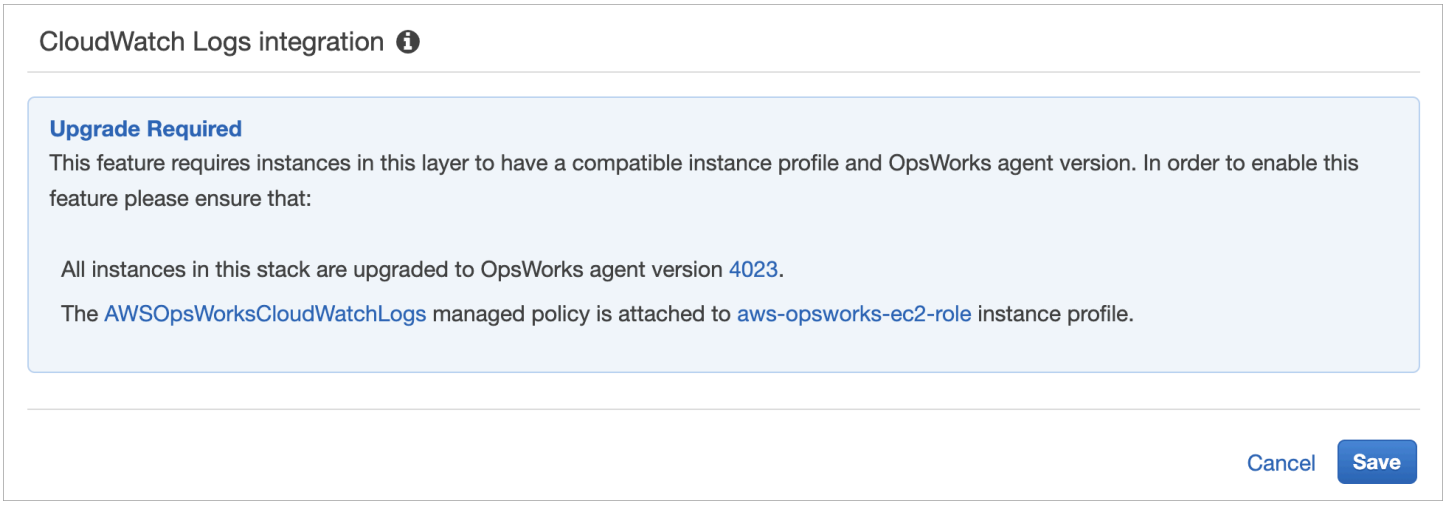
CloudWatch Log memonitor log yang dipilih untuk terjadinya pola yang ditentukan pengguna. Misalnya, Anda dapat memantau log untuk terjadinya istilah literal seperti`NullReferenceException`, atau menghitung jumlah kejadian tersebut. Setelah Anda mengaktifkan CloudWatch Log in AWS OpsWorks Stacks, agen AWS OpsWorks Stacks mengirimkan log ke CloudWatch Log. Untuk informasi selengkapnya tentang CloudWatch Log, lihat Memulai dengan CloudWatch Log.

Prasyarat

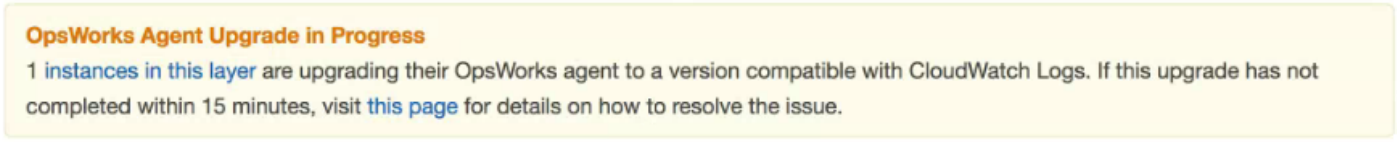
Sebelum Anda dapat mengaktifkan CloudWatch Log, instans Anda harus menjalankan versi 3444 atau yang lebih baru dari agen AWS OpsWorks Stacks di tumpukan Chef 11.10, dan 4023 atau yang lebih baru di tumpukan Chef 12. Anda juga harus menggunakan profil instans yang kompatibel untuk setiap instans yang Anda pantau dengan menggunakan CloudWatch Log.

Jika Anda menggunakan profil instans kustom (yang tidak disediakan AWS OpsWorks Stacks saat Anda membuat tumpukan), AWS OpsWorks Stacks tidak dapat secara otomatis memutakhirkan profil instans. Anda harus secara manual melampirkan AWSOpsWorksCloudWatchLogskebijakan ke profil Anda dengan menggunakan IAM. Untuk selengkapnya, lihat Mengelola kebijakan IAM di Panduan Pengguna IAM.

Jika Anda perlu memutakhirkan versi agen atau profil instans, AWS OpsWorks Stacks menampilkan pengingat yang mirip dengan tangkapan layar berikut saat Anda membuka tab CloudWatch Log di halaman Layer.



Memperbarui agen pada semua instance dalam lapisan dapat memakan waktu. Jika Anda mencoba mengaktifkan CloudWatch Log pada lapisan sebelum upgrade agen selesai, Anda akan melihat pesan yang mirip dengan berikut ini.



Mengaktifkan Log CloudWatch

1. Setelah pemutakhiran profil agen dan instans yang diperlukan selesai, Anda dapat mengaktifkan CloudWatch Log dengan mengatur kontrol slider pada tab CloudWatch Log ke Aktif.

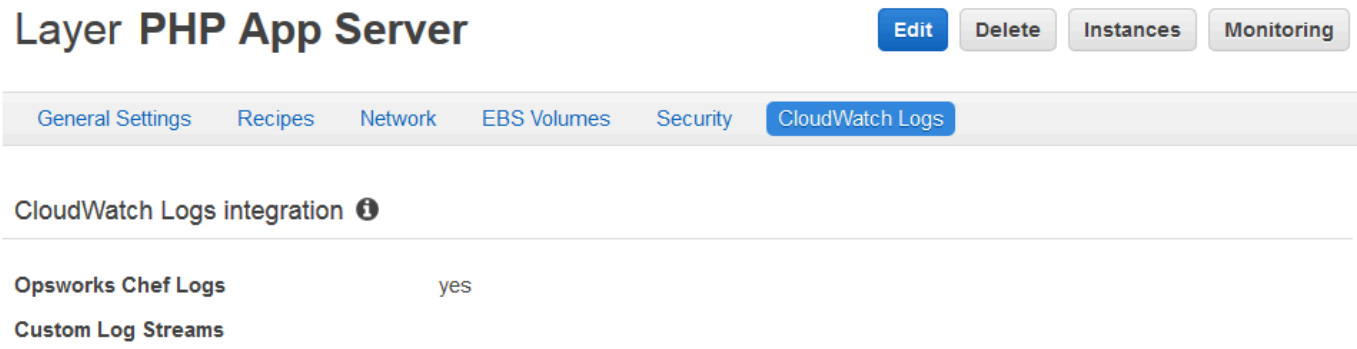


2. Untuk melakukan streaming log perintah, atur slider log perintah Stream ke Aktif. Ini mengirimkan log aktivitas Chef dan perintah yang dimulai pengguna pada instance layer Anda ke Log. CloudWatch

   Data yang disertakan dalam log ini sangat cocok dengan apa yang Anda lihat dalam hasil DescribeCommandsoperasi, saat Anda membuka target URL log. Ini termasuk data tentang`setup`,`configure`,`deploy`,`undeploy`,`start`,`stop`, dan perintah run resep.

3. Untuk mengalirkan log aktivitas yang disimpan di lokasi kustom pada instance layer Anda, misalnya`/var/log/apache/myapp/mylog*`, ketik lokasi kustom di kotak string log kustom Stream, lalu pilih Tambah (+).

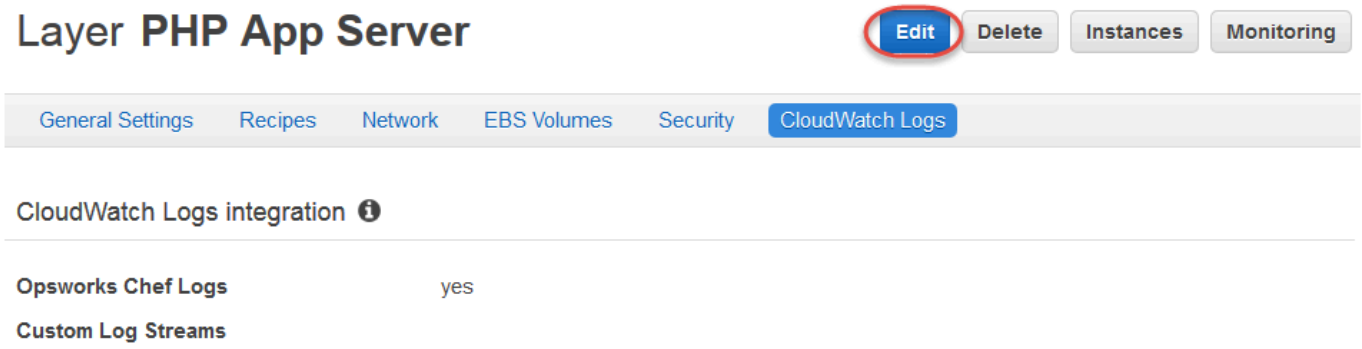
4. Pilih Simpan. Dalam beberapa menit, aliran log AWS OpsWorks Stacks akan terlihat di konsol CloudWatch Log.



Mematikan CloudWatch Log

Untuk mematikan CloudWatch Log, edit pengaturan lapisan Anda.

1. Pada halaman properti layer Anda, pilih Edit.



2. Pada halaman pengeditan, pilih tab CloudWatch Log.
3. Di area CloudWatch Log, matikan log perintah Stream. Pilih X pada log kustom untuk menghapusnya dari aliran log, jika berlaku.
4. Pilih Simpan.

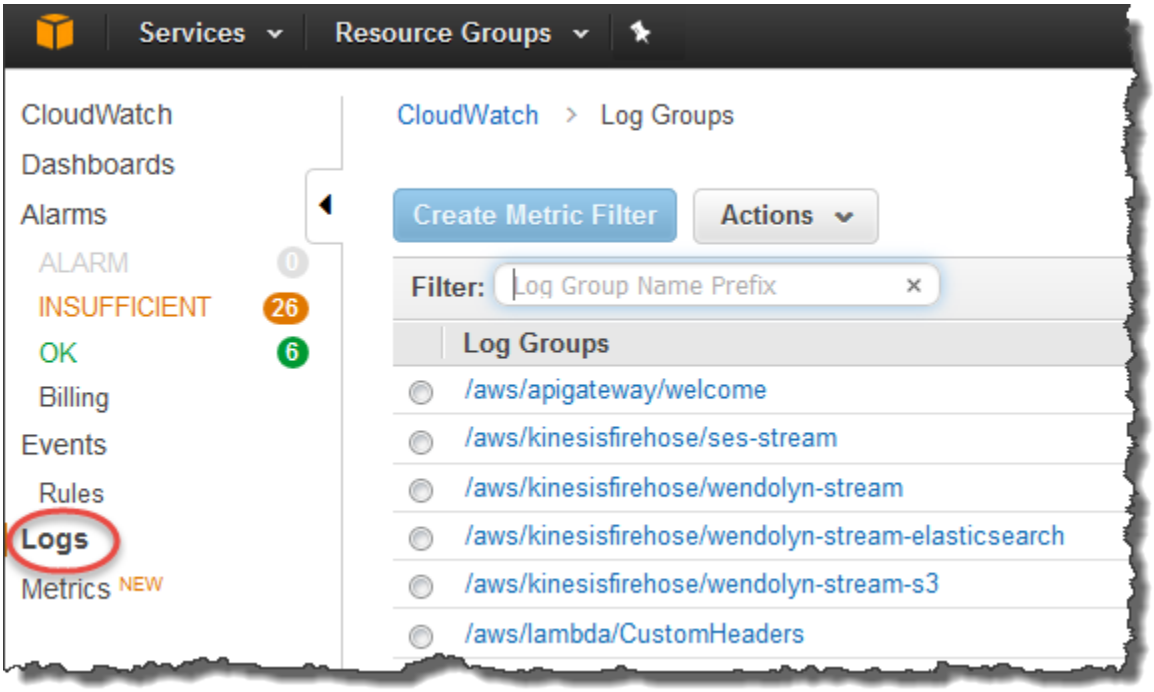
Menghapus Log Streaming dari Log CloudWatch

Setelah Anda menonaktifkan streaming CloudWatch Log dari AWS OpsWorks Tumpukan, log yang ada masih tersedia di konsol manajemen CloudWatch Log. Anda masih dikenakan biaya untuk log yang disimpan, kecuali Anda mengekspor log ke Amazon S3 atau menghapusnya. Untuk informasi selengkapnya tentang mengekspor log ke S3, lihat Mengekspor Data Log ke Amazon S3.

Anda dapat menghapus aliran log dan grup CloudWatch log di konsol manajemen Log, atau dengan menjalankan `delete-log-group` AWS CLI perintah `delete-log-stream`and. Untuk informasi selengkapnya tentang mengubah periode penyimpanan log, lihat Mengubah Penyimpanan Data Log di CloudWatch Log.

Mengelola Log Anda di CloudWatch Log

Log yang Anda streaming dikelola di konsol CloudWatch Log.



AWS OpsWorks membuat grup log default dan aliran log secara otomatis. Grup log untuk data AWS OpsWorks Stacks memiliki nama yang cocok dengan pola berikut:

stack_name/*layer_name*/*chef_log_name*

Log kustom memiliki nama yang cocok dengan pola berikut:

/stack_name/layer_short_name/file_path_name. Nama jalur dibuat lebih mudah dibaca manusia dengan menghilangkan karakter khusus, seperti tanda bintang (*).

Saat Anda menemukan log di CloudWatch Log, Anda dapat mengatur log ke dalam grup, mencari dan memfilter log dengan membuat filter metrik, dan membuat alarm khusus.

Mengkonfigurasi Chef 12.2 Windows Layers untuk Menggunakan Log CloudWatch

CloudWatch Integrasi otomatis log tidak didukung untuk instance berbasis Windows. Tab CloudWatch Log tidak tersedia pada lapisan di tumpukan Chef 12.2. Untuk mengaktifkan streaming secara manual ke CloudWatch Log untuk instance berbasis Windows, lakukan hal berikut.

- Perbarui profil instans untuk instance berbasis Windows sehingga agen CloudWatch Log memiliki izin yang sesuai. Pernyataan AWSOpsWorksCloudWatchLogskebijakan menunjukkan izin mana yang diperlukan.

Biasanya, Anda melakukan tugas ini hanya sekali. Anda kemudian dapat menggunakan profil instance yang diperbarui untuk semua instance Windows dalam satu lapisan.

- Edit file konfigurasi JSON berikut pada setiap instance. File ini mencakup preferensi aliran log, seperti log mana yang akan dipantau.

  ```
  %PROGRAMFILES%\Amazon\Ec2ConfigService\Settings
  \AWS.EC2.Windows.CloudWatch.json
  ```

Anda dapat mengotomatiskan dua tugas sebelumnya dengan membuat resep khusus untuk menangani tugas yang diperlukan dan menetapkannya ke acara Pengaturan lapisan Chef 12.2. Setiap kali Anda memulai instance baru pada layer tersebut, AWS OpsWorks Stacks secara otomatis menjalankan resep Anda setelah instance selesai booting, mengaktifkan Log. CloudWatch

Untuk mematikan CloudWatch Log pada instance berbasis Windows, balikkan prosesnya. Kosongkan kotak centang Aktifkan integrasi CloudWatch Log di kotak dialog Properti EC2 Layanan, hapus preferensi aliran log dari `AWS.EC2.Windows.CloudWatch.json` file; dan berhenti menjalankan resep Chef apa pun yang secara otomatis menetapkan izin CloudWatch Log ke instance baru di lapisan Chef 12.2.

Memantau Tumpukan menggunakan Amazon CloudWatch Events

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat mengonfigurasi aturan di Amazon CloudWatch Events untuk mengingatkan Anda tentang perubahan sumber daya AWS OpsWorks Stacks, dan mengarahkan CloudWatch Acara untuk mengambil tindakan berdasarkan konten peristiwa. Untuk informasi selengkapnya tentang cara memulai CloudWatch Acara dan mengatur aturan, lihat Memulai CloudWatch Acara di Panduan Pengguna CloudWatch Acara.

Jenis acara AWS OpsWorks Stacks berikut didukung di CloudWatch Acara.

Perubahan status instans

Menunjukkan perubahan status instance AWS OpsWorks Stacks.

Perubahan status perintah

Menunjukkan perubahan terjadi dalam keadaan perintah AWS OpsWorks Stacks.

Perubahan status penerapan

Menunjukkan perubahan yang terjadi dalam status penyebaran AWS OpsWorks Stacks.

Peringatan

Menunjukkan kesalahan layanan AWS OpsWorks Stacks dinaikkan.

Untuk informasi selengkapnya tentang jenis peristiwa AWS OpsWorks Tumpukan yang didukung oleh CloudWatch Peristiwa, lihat AWS OpsWorks Menumpuk Peristiwa di Panduan Pengguna CloudWatch Acara.

Keamanan dan Izin

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setiap pengguna Anda harus memiliki AWS kredensi yang sesuai untuk mengakses sumber daya akun Anda. AWS Cara yang disarankan untuk memberikan kredensil kepada pengguna adalah dengan AWS Identity and Access Management(IAM). AWS OpsWorks Stacks terintegrasi dengan IAM untuk memungkinkan Anda mengontrol hal-hal berikut:

- Bagaimana pengguna individu dapat berinteraksi dengan AWS OpsWorks Stacks.

  Misalnya, Anda dapat mengizinkan beberapa pengguna untuk menyebarkan aplikasi ke tumpukan apa pun tetapi tidak memodifikasi tumpukan itu sendiri, sementara memungkinkan pengguna lain mengakses penuh tetapi hanya ke tumpukan tertentu, dan sebagainya.

- Bagaimana AWS OpsWorks Stacks dapat bertindak atas nama Anda untuk mengakses sumber daya tumpukan seperti EC2 instans Amazon dan bucket Amazon S3.

  AWS OpsWorks Stacks menyediakan peran layanan yang memberikan izin untuk tugas-tugas ini.
- Bagaimana aplikasi yang berjalan di EC2 instans Amazon yang dikendalikan oleh AWS OpsWorks Stacks dapat mengakses AWS sumber daya lain, seperti data yang disimpan di bucket Amazon S3.

  Anda dapat menetapkan profil instance ke instance lapisan yang memberikan izin ke aplikasi yang berjalan pada instance tersebut untuk mengakses sumber daya lain. AWS
- Cara mengelola kunci SSH berbasis pengguna dan menggunakan SSH atau RDP untuk terhubung ke instance.

  Untuk setiap tumpukan, pengguna administratif dapat menetapkan setiap pengguna kunci SSH pribadi, atau memberi wewenang kepada pengguna untuk menentukan kunci mereka sendiri. Anda juga dapat mengotorisasi akses SSH atau RDP dan hak istimewa sudo atau administrator pada instance tumpukan untuk setiap pengguna.

Aspek keamanan lainnya adalah sebagai berikut:

- Cara mengelola pembaruan sistem operasi instans Anda dengan tambalan keamanan terbaru.

  Untuk informasi selengkapnya, lihat Mengelola Pembaruan Keamanan.
- Cara mengonfigurasi grup EC2 keamanan Amazon untuk mengontrol lalu lintas jaringan ke dan dari instans Anda.

  Cara menentukan grup keamanan khusus alih-alih grup keamanan default AWS OpsWorks Stacks. Untuk informasi selengkapnya, lihat Menggunakan Grup Keamanan.

Topik

- Mengelola AWS OpsWorks Izin Pengguna Stacks
- Mengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda
- Pencegahan deputi kebingungan lintas layanan di Stacks AWS OpsWorks
- Menentukan Izin untuk Aplikasi yang Berjalan pada instance EC2
- Mengelola Akses SSH
- Mengelola Pembaruan Keamanan Linux

- Menggunakan Grup Keamanan

Mengelola AWS OpsWorks Izin Pengguna Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sebagai praktik terbaik, batasi pengguna AWS OpsWorks Stacks ke serangkaian tindakan tertentu atau kumpulan sumber daya tumpukan. Anda dapat mengontrol izin pengguna AWS OpsWorks Stacks dengan dua cara: dengan menggunakan halaman Izin AWS OpsWorks Tumpukan, dan dengan menerapkan kebijakan IAM yang sesuai.

Halaman OpsWorks Izin—atau tindakan CLI atau API yang setara—memungkinkan Anda mengontrol izin pengguna di lingkungan multipengguna berdasarkan per-tumpukan dengan menetapkan setiap pengguna salah satu dari beberapa tingkat izin. Setiap tingkat memberikan izin untuk serangkaian tindakan standar untuk sumber daya tumpukan tertentu. Dengan menggunakan halaman Izin, Anda dapat mengontrol hal-hal berikut:

- Siapa yang dapat mengakses setiap tumpukan.
- Tindakan mana yang diizinkan dilakukan setiap pengguna di setiap tumpukan.

  Misalnya, Anda dapat mengizinkan beberapa pengguna untuk hanya melihat tumpukan sementara yang lain dapat menyebarkan aplikasi, menambahkan instance, dan sebagainya.
- Siapa yang bisa mengelola setiap tumpukan.

  Anda dapat mendelegasikan pengelolaan setiap tumpukan ke satu atau lebih pengguna tertentu.
- Siapa yang memiliki akses SSH tingkat pengguna dan hak istimewa sudo (Linux) atau akses RDP dan hak administrator (Windows) pada setiap instance Amazon stack. EC2

  Anda dapat memberikan atau menghapus izin ini secara terpisah untuk setiap pengguna kapan saja.

> **Important**
>
> Menolak akses SSH/RDP tidak serta merta mencegah pengguna masuk ke instance. Jika Anda menentukan EC2 key pair Amazon untuk sebuah instance, setiap pengguna dengan kunci pribadi yang sesuai dapat masuk atau menggunakan kunci untuk mengambil kata sandi administrator Windows. Untuk informasi selengkapnya, lihat Mengelola Akses SSH.

Anda dapat menggunakan konsol IAM, CLI, atau API untuk menambahkan kebijakan ke pengguna yang memberikan izin eksplisit untuk AWS OpsWorks berbagai sumber daya dan tindakan Stacks.

- Menggunakan kebijakan IAM untuk menentukan izin lebih fleksibel daripada menggunakan tingkat izin.
- Anda dapat mengatur Identitas IAM (pengguna, grup pengguna, dan peran), yang memberikan izin untuk identitas IAM, seperti pengguna dan grup pengguna, atau menentukan peran yang dapat dikaitkan dengan pengguna federasi.
- Kebijakan IAM adalah satu-satunya cara untuk memberikan izin untuk tindakan AWS OpsWorks Tumpukan kunci tertentu.

  Misalnya, Anda harus menggunakan IAM untuk memberikan izin untuk `opsworks:CreateStack` dan`opsworks:CloneStack`, yang masing-masing digunakan untuk membuat dan mengkloning tumpukan.

Meskipun tidak secara eksplisit mungkin untuk mengimpor pengguna federasi di konsol, pengguna federasi dapat secara implisit membuat profil pengguna dengan memilih Pengaturan Saya di kanan atas konsol AWS OpsWorks Stacks, dan kemudian memilih Pengguna, juga di kanan atas. Pada halaman Pengguna, pengguna federasi—yang akunnya dibuat dengan menggunakan API atau CLI, atau secara implisit melalui konsol—dapat mengelola akun mereka mirip dengan pengguna non-federasi.

Kedua pendekatan tersebut tidak saling eksklusif dan terkadang berguna untuk menggabungkannya; AWS OpsWorks Tumpukan kemudian mengevaluasi kedua set izin. Misalnya, Anda ingin mengizinkan pengguna untuk menambah atau menghapus instance tetapi tidak menambah atau menghapus lapisan. Tak satu pun dari tingkat izin AWS OpsWorks Stacks memberikan set izin tertentu. Namun, Anda dapat menggunakan halaman Izin untuk memberi pengguna tingkat izin Kelola, yang memungkinkan mereka melakukan sebagian besar operasi tumpukan, lalu menerapkan

kebijakan IAM yang menolak izin untuk menambah atau menghapus lapisan. Untuk informasi selengkapnya, lihat Mengontrol akses ke AWS sumber daya menggunakan kebijakan.

Berikut ini adalah model khas untuk mengelola izin pengguna. Dalam setiap kasus, pembaca (Anda) diasumsikan sebagai pengguna administratif.

1. Gunakan konsol IAM untuk menerapkan AWSOpsWorks_FullAccess kebijakan ke satu atau beberapa pengguna administratif.
2. Buat pengguna untuk setiap pengguna nonadministratif dengan kebijakan yang tidak memberikan izin AWS OpsWorks Stacks.

   Jika pengguna hanya memerlukan akses ke AWS OpsWorks Stacks, Anda mungkin tidak perlu menerapkan kebijakan sama sekali. Sebagai gantinya, Anda dapat mengelola izin mereka dengan halaman AWS OpsWorks Izin Tumpukan.
3. Gunakan halaman AWS OpsWorks Stacks Users untuk mengimpor pengguna nonadministratif ke AWS OpsWorks Stacks.
4. Untuk setiap tumpukan, gunakan halaman Izin tumpukan untuk menetapkan tingkat izin untuk setiap pengguna.
5. Jika diperlukan, sesuaikan tingkat izin pengguna dengan menerapkan kebijakan IAM yang dikonfigurasi dengan tepat.

Untuk rekomendasi selengkapnya tentang mengelola pengguna, lihatPraktik Terbaik: Mengelola Izin.

Untuk informasi selengkapnya tentang praktik terbaik IAM, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Topik

- Mengelola AWS OpsWorks Pengguna Stacks
- Memberikan Izin AWS OpsWorks Per-Stack Pengguna Stacks
- Mengelola Izin AWS OpsWorks Tumpukan dengan Melampirkan Kebijakan IAM
- Contoh kebijakan
- AWS OpsWorks Tingkat Izin Tumpukan

Mengelola AWS OpsWorks Pengguna Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sebelum Anda dapat mengimpor pengguna ke AWS OpsWorks Stacks dan memberi mereka izin, Anda harus terlebih dahulu membuat pengguna untuk setiap individu. Untuk membuat pengguna IAM, mulailah dengan masuk AWS sebagai pengguna yang telah diberikan izin yang ditentukan dalam kebijakan IAMFull Access. Anda kemudian menggunakan konsol IAM untuk membuat pengguna IAM untuk semua orang yang perlu mengakses AWS OpsWorks Stacks. Anda kemudian dapat mengimpor pengguna tersebut ke AWS OpsWorks Stacks dan memberikan izin pengguna sebagai berikut:

Pengguna AWS OpsWorks Stacks Reguler

Pengguna reguler tidak memerlukan kebijakan terlampir. Jika mereka memilikinya, biasanya tidak menyertakan izin AWS OpsWorks Stacks apa pun. Sebagai gantinya, gunakan halaman Izin AWS OpsWorks Tumpukan untuk menetapkan salah satu tingkat izin berikut kepada pengguna reguler berdasarkan. stack-by-stack

- Tampilkan izin memungkinkan pengguna untuk melihat tumpukan, tetapi tidak melakukan operasi apa pun.
- Menyebarkan izin termasuk izin Tampilkan dan juga memungkinkan pengguna untuk menyebarkan dan memperbarui aplikasi.
- Mengelola izin termasuk izin Deploy dan juga memungkinkan pengguna untuk melakukan operasi manajemen tumpukan, seperti menambahkan lapisan atau instance, menggunakan halaman Izin untuk mengatur izin pengguna, dan mengaktifkan hak istimewa mereka sendiri. SSH/RDP and sudo/admin
- Tolak izin menolak akses ke tumpukan.

Jika tingkat izin ini tidak sesuai dengan yang Anda inginkan untuk pengguna tertentu, Anda dapat menyesuaikan izin pengguna dengan menerapkan kebijakan IAM. Misalnya, Anda mungkin ingin

menggunakan halaman Izin AWS OpsWorks Tumpukan untuk menetapkan tingkat izin Kelola ke pengguna, yang memberi mereka izin untuk melakukan semua operasi manajemen tumpukan, tetapi tidak untuk membuat atau mengkloning tumpukan. Anda kemudian dapat menerapkan kebijakan yang membatasi izin tersebut dengan menolak izin mereka untuk menambah atau menghapus lapisan atau menambah izin tersebut dengan mengizinkan mereka membuat atau mengkloning tumpukan. Untuk informasi selengkapnya, lihat Mengelola Izin AWS OpsWorks Tumpukan dengan Melampirkan Kebijakan IAM.

AWS OpsWorks Tumpukan Pengguna Administratif

Pengguna administratif adalah pemilik akun atau pengguna IAM dengan izin yang ditentukan oleh kebijakan. AWSOpsWorks_FullAccess Selain izin yang diberikan kepada Kelola pengguna, kebijakan ini mencakup izin untuk tindakan yang tidak dapat diberikan melalui halaman Izin, seperti berikut:

- Mengimpor pengguna ke Stacks AWS OpsWorks
- Membuat dan mengkloning tumpukan

Untuk kebijakan selengkapnya, lihatContoh kebijakan. Untuk daftar izin terperinci yang hanya dapat diberikan kepada pengguna dengan menerapkan kebijakan IAM, lihat. AWS OpsWorks Tingkat Izin Tumpukan

Topik

- Pengguna dan Wilayah
- Membuat Pengguna Administratif AWS OpsWorks Stacks
- Membuat pengguna IAM untuk Stacks AWS OpsWorks
- Mengimpor Pengguna ke Tumpukan AWS OpsWorks
- Mengedit AWS OpsWorks Pengaturan Pengguna Stacks

Pengguna dan Wilayah

AWS OpsWorks Pengguna tumpukan tersedia dalam titik akhir regional tempat mereka dibuat. Anda dapat membuat pengguna di salah satu Wilayah berikut.

- Wilayah AS Timur (Ohio)
- Wilayah AS Timur (Virginia Utara)
- Wilayah AS Barat (Oregon)

- Wilayah Barat AS (California U.)
- Wilayah Kanada (Tengah) (hanya API; tidak tersedia di AWS Management Console
- Wilayah Asia Pasifik (Mumbai)
- Wilayah Asia Pasifik (Singapura)
- Wilayah Asia Pasifik (Sydney)
- Wilayah Asia Pacific (Tokyo)
- Wilayah Asia Pasifik (Seoul)
- Wilayah Eropa (Frankfurt)
- Wilayah Eropa (Irlandia)
- Wilayah Eropa (London)
- Wilayah Eropa (Paris)
- Wilayah Amerika Selatan (Sao Paulo)

Saat Anda mengimpor pengguna ke AWS OpsWorks Stacks, Anda mengimpornya ke salah satu titik akhir regional; jika Anda ingin pengguna tersedia di lebih dari satu Wilayah, Anda harus mengimpor pengguna ke Wilayah tersebut. Anda juga dapat mengimpor pengguna AWS OpsWorks Stacks dari satu Wilayah ke Wilayah lain; jika Anda mengimpor pengguna ke Wilayah yang sudah memiliki pengguna dengan nama yang sama, pengguna yang diimpor akan menggantikan pengguna yang sudah ada. Untuk informasi selengkapnya tentang mengimpor pengguna, lihatMengimpor Pengguna.

Membuat Pengguna Administratif AWS OpsWorks Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat membuat pengguna administratif AWS OpsWorks Stacks dengan menambahkan `AWSOpsWorks_FullAccess` kebijakan ke pengguna, yang memberi pengguna tersebut izin AWS OpsWorks Menumpuk Akses Penuh. Untuk informasi selengkapnya tentang membuat pengguna administratif, lihat Membuat pengguna administratif.

> **Note**
>
> AWSOpsWorks_FullAccess Kebijakan ini memungkinkan pengguna untuk membuat dan mengelola AWS OpsWorks tumpukan Stacks, tetapi pengguna tidak dapat membuat peran layanan IAM untuk tumpukan; mereka harus menggunakan peran yang ada. Pengguna pertama yang membuat tumpukan harus memiliki izin IAM tambahan, seperti yang dijelaskan dalam. Izin Administratif Saat pengguna ini membuat tumpukan pertama, AWS OpsWorks Stacks membuat peran layanan IAM dengan izin yang diperlukan. Setelah itu, setiap pengguna dengan `opsworks:CreateStack` izin dapat menggunakan peran itu untuk membuat tumpukan tambahan. Untuk informasi selengkapnya, lihat Mengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda.

Saat membuat pengguna, Anda dapat menambahkan kebijakan tambahan yang dikelola pelanggan untuk menyempurnakan izin pengguna, sesuai kebutuhan. Misalnya, Anda mungkin ingin pengguna administratif dapat membuat atau menghapus tumpukan, tetapi tidak mengimpor pengguna baru. Untuk informasi selengkapnya, lihat Mengelola Izin AWS OpsWorks Tumpukan dengan Melampirkan Kebijakan IAM.

Jika Anda memiliki beberapa pengguna administratif, alih-alih menyetel izin secara terpisah untuk setiap pengguna, Anda dapat menambahkan AWSOpsWorks_FullAccess kebijakan ke grup IAM dan menambahkan pengguna ke grup tersebut.

Untuk informasi tentang membuat grup, lihat Membuat grup pengguna IAM. Saat Anda membuat grup, tambahkan AWSOpsWorks_FullAccesskebijakan. Anda juga dapat menambahkan AdministratorAccesskebijakan, yang mencakup AWSOpsWorks_FullAccessizin.

Untuk informasi tentang menambahkan izin ke grup yang ada, lihat Melampirkan kebijakan ke grup pengguna IAM.

Membuat pengguna IAM untuk Stacks AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sebelum Anda dapat mengimpor pengguna IAM ke AWS OpsWorks Stacks, Anda harus membuatnya. Anda dapat melakukan ini menggunakan konsol IAM, baris perintah, atau API. Untuk petunjuk selengkapnya, lihat Membuat pengguna IAM di AWS akun Anda.

Perhatikan bahwa tidak seperti pengguna administratif, Anda tidak perlu melampirkan kebijakan untuk menentukan izin. Anda dapat mengatur izin setelah mengimpor pengguna ke AWS OpsWorks Stacks, seperti yang dijelaskan di. Mengelola izin

Untuk informasi selengkapnya tentang membuat pengguna dan grup IAM, lihat Memulai IAM.

Mengimpor Pengguna ke Tumpukan AWS OpsWorks

Important

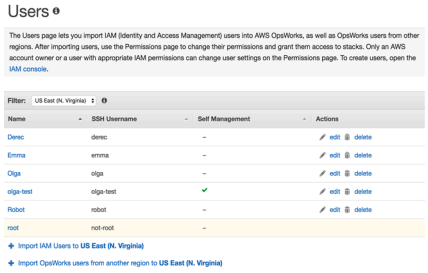
AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pengguna administratif dapat mengimpor pengguna ke AWS OpsWorks Stacks; mereka juga dapat mengimpor pengguna AWS OpsWorks Stacks dari satu titik akhir regional ke titik akhir lainnya. Saat Anda mengimpor pengguna ke AWS OpsWorks Stacks, Anda mengimpornya ke salah satu titik akhir regional AWS OpsWorks Stacks. Jika Anda ingin pengguna tersedia di lebih dari satu Wilayah, Anda harus mengimpor pengguna ke Wilayah tersebut.

Meskipun tidak secara eksplisit mungkin untuk mengimpor pengguna federasi di konsol, pengguna federasi dapat secara implisit membuat profil pengguna dengan memilih Pengaturan Saya di kanan atas konsol AWS OpsWorks Stacks, dan kemudian memilih Pengguna, juga di kanan atas. Pada halaman Pengguna, pengguna federasi—yang akunnya dibuat dengan menggunakan API atau CLI, atau secara implisit melalui konsol—dapat mengelola akun mereka mirip dengan pengguna non-federasi.

Untuk mengimpor pengguna ke AWS OpsWorks Stacks

1. Masuk ke AWS OpsWorks Stacks sebagai pengguna administratif atau sebagai pemilik akun.
2. Pilih Pengguna di kanan atas untuk membuka halaman Pengguna.



3. Pilih Impor Pengguna IAM ke **<** ***region name*** **>** untuk menampilkan pengguna yang tersedia, tetapi yang belum diimpor.



4. Isi kotak centang Pilih semua, atau pilih satu atau beberapa pengguna individu. Setelah selesai, pilih Impor ke OpsWorks.

> **Note**
>
> Setelah Anda mengimpor pengguna ke AWS OpsWorks Stacks, jika Anda menggunakan konsol IAM atau API untuk menghapus pengguna dari akun Anda, pengguna tidak secara otomatis kehilangan akses SSH yang telah Anda berikan melalui Stacks. AWS OpsWorks Anda juga harus menghapus pengguna dari AWS OpsWorks Tumpukan dengan membuka halaman Pengguna, dan memilih hapus di kolom Tindakan pengguna.

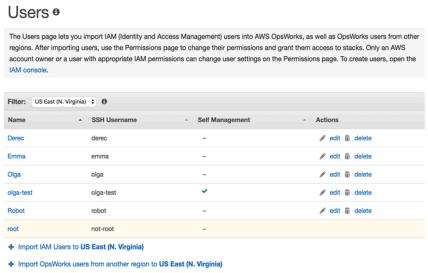
Untuk mengimpor pengguna AWS OpsWorks Stacks dari satu Wilayah ke Wilayah lainnya

AWS OpsWorks Pengguna tumpukan tersedia dalam titik akhir regional tempat mereka dibuat. Anda dapat membuat pengguna di Wilayah yang ditampilkan diPengguna dan Wilayah.

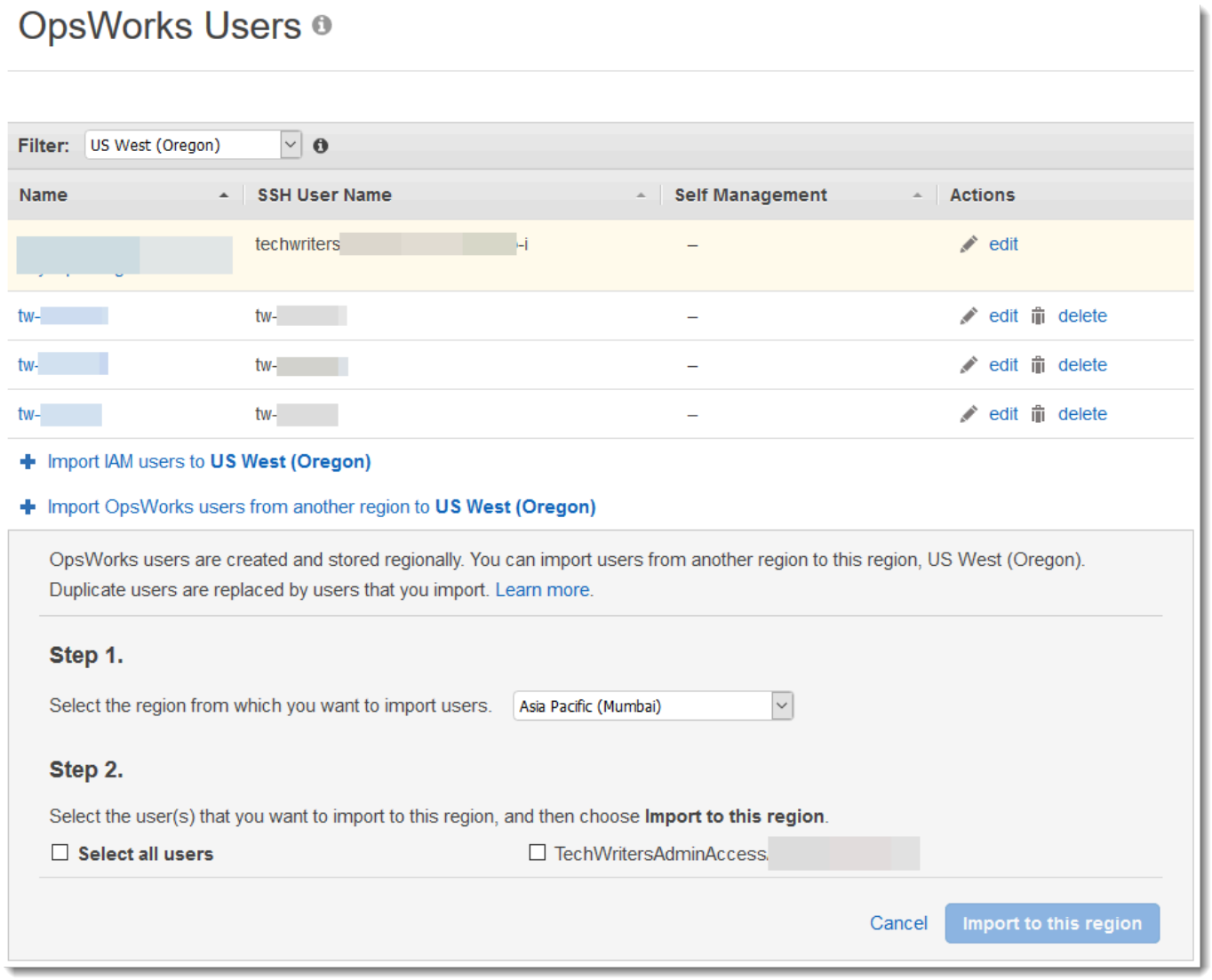
Anda dapat mengimpor pengguna AWS OpsWorks Tumpukan dari satu Wilayah ke Wilayah tempat daftar Pengguna Anda saat ini difilter. Jika Anda mengimpor pengguna ke Wilayah yang sudah memiliki pengguna dengan nama yang sama, pengguna yang diimpor akan menggantikan pengguna yang sudah ada.

1. Masuk ke AWS OpsWorks Stacks sebagai pengguna administratif atau sebagai pemilik akun.

2. Pilih Pengguna di kanan atas untuk membuka halaman Pengguna. Jika Anda memiliki pengguna AWS OpsWorks Stacks di lebih dari satu Wilayah, gunakan kontrol Filter untuk memfilter Wilayah yang ingin Anda impor pengguna.



3. Pilih Impor AWS OpsWorks Tumpukan pengguna dari Wilayah lain ke < ***current region*** >.



4. Pilih Wilayah tempat Anda ingin mengimpor pengguna AWS OpsWorks Stacks.

5. Pilih satu atau beberapa pengguna yang akan diimpor, atau pilih semua pengguna, lalu pilih Impor ke wilayah ini. Tunggu AWS OpsWorks Stacks untuk menampilkan pengguna yang diimpor dalam daftar Pengguna.

Unix IDs dan Pengguna Membuat Tumpukan Luar AWS OpsWorks

AWS OpsWorks menetapkan pengguna pada AWS OpsWorks Stacks instance Unix ID (UID) nilai antara 2000 dan 4000. Karena AWS OpsWorks cadangan kisaran 2000-4000 UIDs, pengguna yang Anda buat di luar AWS OpsWorks (dengan menggunakan resep buku masak, atau dengan mengimpor pengguna ke AWS OpsWorks dari IAM, misalnya) dapat memiliki UIDs yang ditimpa oleh Stacks untuk pengguna lain. AWS OpsWorks Hal ini dapat mengakibatkan pengguna yang telah Anda buat di luar AWS OpsWorks Stacks tidak muncul di hasil pencarian tas data, atau dikecualikan dari operasi bawaan AWS OpsWorks `sync_remote_users` Stacks.

Proses eksternal juga dapat membuat pengguna dengan AWS OpsWorks Stacks UIDs yang dapat menimpa. Beberapa paket sistem operasi, misalnya, dapat membuat pengguna sebagai bagian dari proses pasca-instalasi. Ketika Anda atau proses perangkat lunak membuat pengguna pada sistem operasi berbasis Linux tanpa secara eksplisit menentukan UID — yang merupakan default — UID yang ditetapkan oleh Stacks adalah + 1. AWS OpsWorks *`<highest existing AWS OpsWorks UID>`*

Sebagai praktik terbaik, buat pengguna AWS OpsWorks Stacks dan kelola aksesnya di konsol AWS OpsWorks Stacks AWS CLI, atau dengan menggunakan SDK. AWS Jika Anda membuat pengguna pada instance AWS OpsWorks Stacks di luar AWS OpsWorks, gunakan *`UnixID`* nilai yang lebih besar dari 4000.

Mengedit AWS OpsWorks Pengaturan Pengguna Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda mengimpor pengguna, Anda dapat mengedit pengaturan mereka, sebagai berikut:

Untuk mengedit setelan pengguna

1. Pada halaman Pengguna, pilih edit di kolom Tindakan pengguna.
2. Anda dapat menentukan pengaturan berikut.

   Manajemen Diri

   Pilih Ya untuk memungkinkan pengguna menggunakan MySettings halaman untuk menentukan kunci SSH pribadinya.

   > **Note**
   >
   > Anda juga dapat mengaktifkan pengelolaan mandiri dengan menambahkan kebijakan IAM ke identitas IAM yang memberikan izin untuk tindakan dan tindakan. DescribeMyUserProfileUpdateMyUserProfile

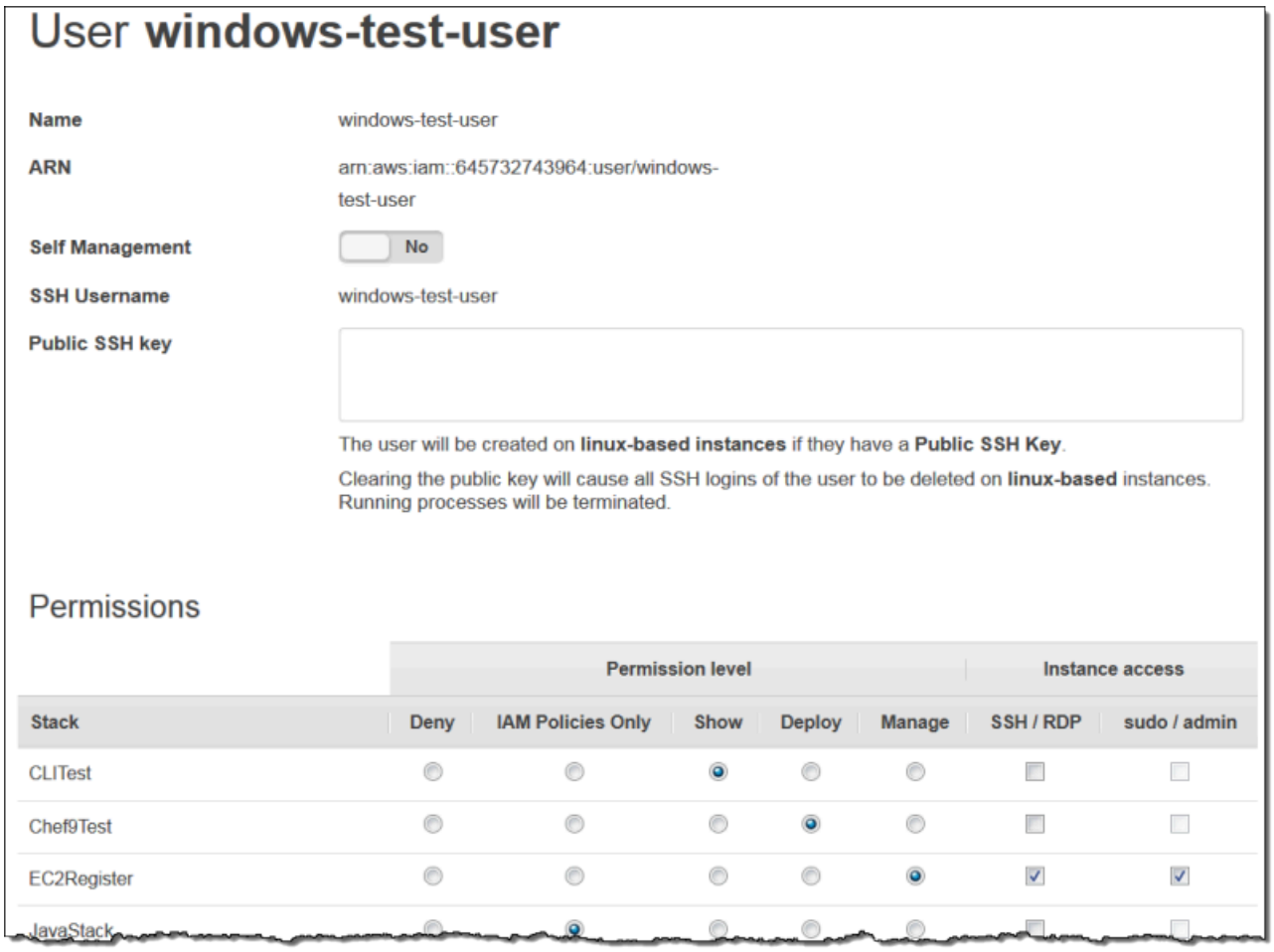
   Kunci SSH publik

   (Opsional) Masukkan kunci SSH publik untuk pengguna. Kunci ini akan muncul di halaman Pengaturan Saya pengguna. Jika Anda mengaktifkan manajemen diri, pengguna dapat mengedit Pengaturan Saya dan menentukan kuncinya sendiri. Untuk informasi selengkapnya, lihat Mendaftarkan Kunci SSH Publik Pengguna.

   AWS OpsWorks Stacks menginstal kunci ini di semua instance Linux; pengguna dapat menggunakan kunci pribadi terkait untuk masuk. Untuk informasi selengkapnya, lihat Login dengan SSH. Anda tidak dapat menggunakan kunci ini dengan tumpukan Windows.

   Izin

   (Opsional) Tetapkan tingkat izin pengguna untuk setiap tumpukan di satu tempat alih-alih mengaturnya secara terpisah dengan menggunakan halaman Izin setiap tumpukan. Untuk informasi selengkapnya tentang tingkat izin, lihatMemberikan Izin Per-Stack.



Memberikan Izin AWS OpsWorks Per-Stack Pengguna Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Cara termudah untuk mengelola izin pengguna AWS OpsWorks Stacks adalah dengan menggunakan halaman Izin tumpukan. Setiap tumpukan memiliki halamannya sendiri, yang memberikan izin untuk tumpukan itu.

Anda harus masuk sebagai pengguna administratif atau Kelola pengguna untuk mengubah pengaturan izin apa pun. Daftar ini hanya menampilkan pengguna yang telah diimpor ke AWS OpsWorks Stacks. Untuk informasi tentang cara membuat dan mengimpor pengguna, lihatMengelola Pengguna.

Tingkat izin default adalah Kebijakan IAM Saja, yang hanya memberi pengguna izin yang ada dalam kebijakan IAM mereka.

- Saat Anda mengimpor pengguna dari IAM atau dari Wilayah lain, pengguna akan ditambahkan ke daftar untuk semua tumpukan yang ada dengan tingkat izin Khusus Kebijakan IAM.
- Secara default, pengguna yang baru saja Anda impor dari Wilayah lain tidak memiliki akses ke tumpukan di Wilayah tujuan. Jika Anda mengimpor pengguna dari Wilayah lain, agar mereka dapat mengelola tumpukan di Wilayah tujuan, mereka harus diberi izin ke tumpukan tersebut setelah Anda mengimpor pengguna.
- Saat Anda membuat tumpukan baru, semua pengguna saat ini ditambahkan ke daftar dengan tingkat izin Hanya Kebijakan IAM.

Topik

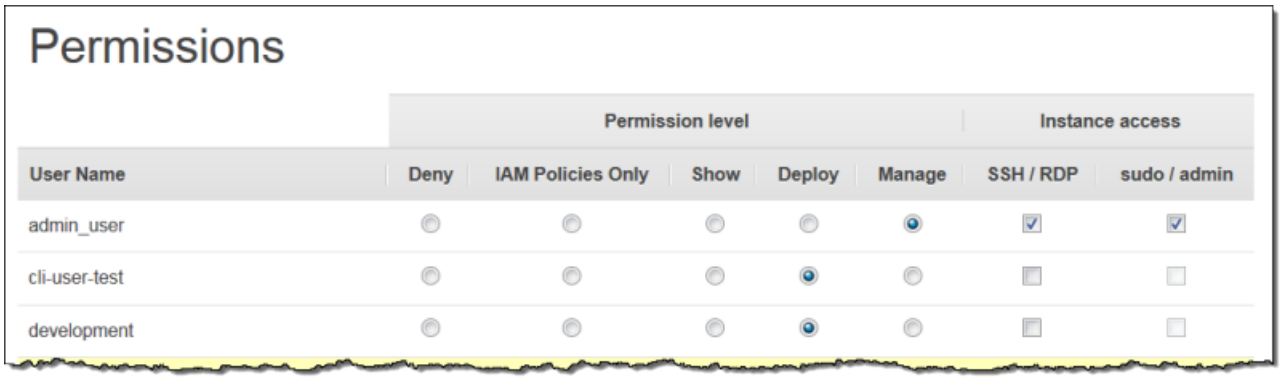
- Mengatur Izin Pengguna
- Melihat Izin Anda
- Menggunakan Kunci Kondisi IAM untuk Memverifikasi Kredensi Sementara

Mengatur Izin Pengguna

Untuk menyetel izin pengguna

1. Di panel navigasi, pilih Izin.
2. Pada halaman Izin, pilih Edit.
3. Ubah pengaturan tingkat Izin dan akses Instance:
    - Gunakan pengaturan tingkat Izin untuk menetapkan salah satu tingkat izin standar untuk setiap pengguna, yang menentukan apakah pengguna dapat mengakses tumpukan ini dan tindakan apa yang dapat dilakukan pengguna. Jika pengguna memiliki kebijakan IAM, AWS OpsWorks Stacks mengevaluasi kedua set izin. Sebagai contoh lihatContoh kebijakan.
    - Pengaturan Instance access SSH/RDP menentukan apakah pengguna memiliki akses SSH (Linux) atau RDP (Windows) ke instance stack.

Jika Anda mengotorisasi akses SSH/RDP, Anda dapat memilih sudo/admin secara opsional, yang memberikan hak istimewa sudo (Linux) atau administratif (Windows) pengguna pada instance tumpukan.



Anda dapat menetapkan setiap pengguna ke salah satu tingkat izin berikut. Untuk daftar tindakan yang diizinkan oleh setiap level, lihatAWS OpsWorks Tingkat Izin Tumpukan.

Menyangkal

Pengguna tidak dapat melakukan tindakan AWS OpsWorks Stacks apa pun di tumpukan, meskipun mereka memiliki kebijakan IAM yang memberikan izin akses penuh kepada AWS OpsWorks Stacks. Anda dapat menggunakan ini, misalnya, untuk menolak akses beberapa pengguna ke tumpukan untuk produk yang belum dirilis.

Hanya Kebijakan IAM

Tingkat default, yang ditetapkan untuk semua pengguna yang baru diimpor, dan untuk semua pengguna untuk tumpukan yang baru dibuat. Izin pengguna ditentukan oleh kebijakan IAM mereka. Jika pengguna tidak memiliki kebijakan IAM, atau kebijakan mereka tidak memiliki izin AWS OpsWorks Tumpukan eksplisit, mereka tidak dapat mengakses tumpukan. Pengguna administratif biasanya diberi level ini karena kebijakan IAM mereka sudah memberikan izin akses penuh.

Tampilkan

Pengguna dapat melihat tumpukan, tetapi tidak melakukan operasi apa pun. Misalnya, manajer mungkin ingin memantau tumpukan akun, tetapi tidak perlu menerapkan aplikasi atau memodifikasi tumpukan dengan cara apa pun.

Menyebarkan

Termasuk izin Tampilkan dan juga memungkinkan pengguna untuk menyebarkan aplikasi. Misalnya, pengembang aplikasi mungkin perlu menerapkan pembaruan ke instance tumpukan tetapi tidak menambahkan lapisan atau instance ke tumpukan.

Mengelola

Termasuk izin Deploy dan juga memungkinkan pengguna untuk melakukan berbagai operasi manajemen tumpukan, termasuk:

- Menambahkan atau menghapus layer dan instance.
- Menggunakan halaman Izin tumpukan untuk menetapkan tingkat izin kepada pengguna.
- Mendaftarkan atau membatalkan pendaftaran sumber daya.

Misalnya, setiap tumpukan dapat memiliki manajer yang ditunjuk yang bertanggung jawab untuk memastikan bahwa tumpukan memiliki jumlah dan jenis instance yang sesuai, menangani pembaruan paket dan sistem operasi, dan sebagainya.

> **Note**
>
> Level Kelola tidak memungkinkan pengguna membuat atau mengkloning tumpukan. Izin tersebut harus diberikan oleh kebijakan IAM. Sebagai contoh, lihat Kelola Izin.

Jika pengguna juga memiliki kebijakan IAM, AWS OpsWorks Stacks mengevaluasi kedua set izin. Ini memungkinkan Anda menetapkan tingkat izin kepada pengguna dan kemudian menerapkan kebijakan untuk membatasi atau menambah tindakan yang diizinkan pada level tersebut. Misalnya, Anda dapat menerapkan kebijakan yang memungkinkan pengguna Kelola membuat atau mengkloning tumpukan, atau menyangkal kemampuan pengguna tersebut untuk mendaftarkan atau membatalkan pendaftaran sumber daya. Untuk beberapa contoh kebijakan semacam itu, lihatContoh kebijakan.

> **Note**
>
> Jika kebijakan pengguna mengizinkan tindakan tambahan, hasilnya akan muncul untuk mengganti setelan halaman Izin. Misalnya, jika pengguna memiliki kebijakan yang mengizinkan CreateLayertindakan tetapi Anda menggunakan halaman Izin untuk menentukan izin Penerapan, pengguna tetap diizinkan untuk membuat lapisan. Pengecualian

untuk aturan ini adalah opsi Tolak, yang menolak akses tumpukan bahkan ke pengguna dengan AWSOpsWorks_FullAccess kebijakan. Untuk informasi selengkapnya, lihat Mengontrol akses ke AWS sumber daya menggunakan kebijakan.

Melihat Izin Anda

Jika manajemen mandiri diaktifkan, pengguna dapat melihat ringkasan tingkat izin mereka untuk setiap tumpukan dengan memilih Pengaturan Saya, di kanan atas. Pengguna juga dapat mengakses Pengaturan Saya jika kebijakan mereka memberikan izin untuk tindakan DescribeMyUserProfiledan UpdateMyUserProfiletindakan.

Menggunakan Kunci Kondisi IAM untuk Memverifikasi Kredensi Sementara

AWS OpsWorks Stacks memiliki lapisan otorisasi bawaan yang mendukung kasus otorisasi tambahan (seperti pengelolaan akses read-only atau read-write yang disederhanakan ke tumpukan untuk pengguna individu). Lapisan otorisasi ini bergantung pada penggunaan kredensi sementara. Karena itu, Anda tidak dapat menggunakan `aws:TokenIssueTime` kondisi untuk memverifikasi bahwa pengguna menggunakan kredensi jangka panjang, atau memblokir tindakan dari pengguna yang menggunakan kredensial sementara, seperti yang dijelaskan dalam referensi elemen kebijakan IAM JSON dalam dokumentasi IAM.

Mengelola Izin AWS OpsWorks Tumpukan dengan Melampirkan Kebijakan IAM

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menentukan izin AWS OpsWorks Tumpukan pengguna dengan melampirkan kebijakan IAM. Kebijakan terlampir diperlukan untuk beberapa izin:

- Izin pengguna administratif, seperti mengimpor pengguna.
- Izin untuk beberapa tindakan, seperti membuat atau mengkloning tumpukan.

Untuk daftar lengkap tindakan yang memerlukan kebijakan terlampir, lihatAWS OpsWorks Tingkat Izin Tumpukan.

Anda juga dapat menggunakan kebijakan untuk menyesuaikan tingkat izin yang diberikan melalui halaman Izin. Bagian ini memberikan ringkasan singkat tentang cara menerapkan kebijakan IAM kepada pengguna untuk menentukan izin AWS OpsWorks Stacks. Untuk informasi selengkapnya, lihat Manajemen akses untuk AWS sumber daya.

Kebijakan IAM adalah objek JSON yang berisi satu atau lebih pernyataan. Setiap elemen pernyataan memiliki daftar izin, yang memiliki tiga elemen dasar mereka sendiri:

Tindakan

Tindakan yang dipengaruhi oleh izin. Anda menentukan tindakan AWS OpsWorks Stacks sebagaiopsworks:*action*. An `Action` dapat diatur ke tindakan tertentu sepertiopsworks:`CreateStack`, yang menentukan apakah pengguna diizinkan untuk memanggil `CreateStack`. Anda juga dapat menggunakan wildcard untuk menentukan grup tindakan. Misalnya, `opsworks:Create*` menentukan semua tindakan penciptaan. Untuk daftar lengkap tindakan AWS OpsWorks Stacks, lihat Referensi API AWS OpsWorks Stacks.

Efek

Apakah tindakan yang ditentukan diizinkan atau ditolak.

Sumber Daya

Sumber AWS daya yang dipengaruhi oleh izin. AWS OpsWorks Stacks memiliki satu jenis sumber daya, tumpukan. Untuk menentukan izin untuk sumber daya tumpukan tertentu, atur `Resource` ke ARN tumpukan, yang memiliki format berikut:. `arn:aws:opsworks:`*region*`:`*account_id*`:stack/`*stack_id*`/`

Anda juga dapat menggunakan wildcard. Misalnya, menyetel `Resource` untuk * memberikan izin untuk setiap sumber daya.

Misalnya, kebijakan berikut menyangkal kemampuan pengguna untuk menghentikan instance pada tumpukan yang ID-nya. `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/"
    }
  ]
}
```

Untuk informasi tentang menambahkan izin ke pengguna IAM, lihat. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_change-permissions.html#users_change_permissions-add-console

Untuk informasi selengkapnya tentang cara membuat atau mengubah kebijakan IAM, lihat Kebijakan dan izin di IAM. Untuk beberapa contoh kebijakan AWS OpsWorks Stacks, lihatContoh kebijakan.

Contoh kebijakan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan contoh kebijakan IAM yang dapat diterapkan pada pengguna AWS OpsWorks Stacks.

- Izin Administratifmenjelaskan kebijakan yang digunakan untuk memberikan izin kepada pengguna administratif.
- Kelola Izindan Menyebarkan Izin tunjukkan contoh kebijakan yang dapat diterapkan pada pengguna untuk menambah atau membatasi tingkat izin Kelola dan Penerapan.

  AWS OpsWorks Tumpukan menentukan izin pengguna dengan mengevaluasi izin yang diberikan oleh kebijakan IAM serta izin yang diberikan oleh halaman Izin. Untuk informasi selengkapnya, lihat Mengontrol akses ke AWS sumber daya menggunakan kebijakan. Untuk informasi selengkapnya tentang izin halaman Izin, lihat. AWS OpsWorks Tingkat Izin Tumpukan

Izin Administratif

Gunakan konsol IAM https://console.aws.amazon.com/iam/, untuk mengakses AWSOpsWorks_FullAccess kebijakan, Lampirkan kebijakan ini ke pengguna untuk memberi mereka izin untuk melakukan semua tindakan AWS OpsWorks Tumpukan. Izin IAM diperlukan, antara lain, untuk memungkinkan pengguna administratif mengimpor pengguna.

Anda harus membuat peran IAM yang memungkinkan AWS OpsWorks Stacks bertindak atas nama Anda untuk mengakses AWS sumber daya lain, seperti instans Amazon EC2 . Anda biasanya menangani tugas ini dengan meminta pengguna administratif membuat tumpukan pertama, dan membiarkan AWS OpsWorks Stacks membuat peran untuk Anda. Anda kemudian dapat menggunakan peran itu untuk semua tumpukan berikutnya. Untuk informasi selengkapnya, lihat Mengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda.

Pengguna administratif yang membuat tumpukan pertama harus memiliki izin untuk beberapa tindakan IAM yang tidak disertakan dalam kebijakan. AWSOpsWorks_FullAccess Tambahkan izin berikut ke `Actions` bagian kebijakan. Untuk sintaks JSON yang tepat, pastikan untuk menambahkan koma di antara tindakan dan menghapus tanda koma di akhir daftar tindakan.

```
"iam:PutRolePolicy",
"iam:AddRoleToInstanceProfile",
"iam:CreateInstanceProfile",
"iam:CreateRole"
```

Kelola Izin

Tingkat izin Kelola memungkinkan pengguna untuk melakukan berbagai tindakan manajemen tumpukan, termasuk menambahkan atau menghapus lapisan. Topik ini menjelaskan beberapa kebijakan yang dapat Anda gunakan untuk Mengelola pengguna untuk menambah atau membatasi izin standar.

Tolak Kelola pengguna kemampuan untuk menambah atau menghapus lapisan

Anda dapat membatasi tingkat Kelola izin untuk memungkinkan pengguna melakukan semua tindakan Kelola kecuali menambahkan atau menghapus lapisan dengan menggunakan kebijakan IAM berikut. Ganti*`region`*,*`account_id`*, dan *`stack_id`* dengan nilai yang sesuai dengan konfigurasi Anda.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:CreateLayer",
        "opsworks:DeleteLayer"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

Izinkan pengguna Kelola untuk membuat atau mengkloning tumpukan

Tingkat Kelola izin tidak memungkinkan pengguna untuk membuat atau mengkloning tumpukan. Anda dapat mengubah izin Kelola untuk memungkinkan pengguna membuat atau mengkloning tumpukan dengan menerapkan kebijakan IAM berikut. Ganti *region* dan *account_id* dengan nilai yang sesuai dengan konfigurasi Anda.

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

Tolak Kelola pengguna kemampuan untuk mendaftar atau membatalkan pendaftaran sumber daya

Tingkat izin Kelola memungkinkan pengguna untuk mendaftarkan dan membatalkan pendaftaran Amazon EBS dan sumber daya alamat IP Elastis dengan tumpukan. Anda dapat membatasi izin Kelola agar pengguna dapat melakukan semua tindakan Kelola kecuali mendaftarkan sumber daya dengan menerapkan kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "opsworks:RegisterVolume",
        "opsworks:RegisterElasticIp"
      ],
      "Resource": "*"
    }
  ]
}
```

Izinkan pengguna Kelola untuk mengimpor pengguna

Tingkat Kelola izin tidak memungkinkan pengguna untuk mengimpor pengguna ke AWS OpsWorks Stacks. Anda dapat menambah izin Kelola untuk memungkinkan pengguna mengimpor dan menghapus pengguna dengan menerapkan kebijakan IAM berikut. Ganti *`region`* dan *`account_id`* dengan nilai yang sesuai dengan konfigurasi Anda.

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

Menyebarkan Izin

Tingkat izin Deploy tidak memungkinkan pengguna untuk membuat atau menghapus aplikasi. Anda dapat menambah izin Deploy untuk memungkinkan pengguna membuat dan menghapus aplikasi dengan menerapkan kebijakan IAM berikut. Ganti*region*,*account_id*, dan *stack_id* dengan nilai yang sesuai dengan konfigurasi Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "opsworks:CreateApp",
        "opsworks:DeleteApp"
      ],
      "Resource": "arn:aws:opsworks:region:account_id:stack/stack_id/"
    }
  ]
}
```

AWS OpsWorks Tingkat Izin Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini mencantumkan tindakan yang diizinkan oleh tingkat izin Tampilkan, Terapkan, dan Kelola di halaman Izin AWS OpsWorks Tumpukan. Ini juga mencakup daftar tindakan yang dapat Anda berikan izin hanya dengan menerapkan kebijakan IAM kepada pengguna.

Tampilkan

Level Tampilkan memungkinkan `Describe`*XYZ* perintah, dengan pengecualian berikut:

```
DescribePermissions
DescribeUserProfiles
DescribeMyUserProfile
DescribeStackProvisioningParameters
```

Jika pengguna administratif telah mengaktifkan manajemen mandiri untuk pengguna, Tampilkan pengguna juga dapat menggunakan `DescribeMyUserProfile` dan`UpdateMyUserProfile`. Untuk informasi lebih lanjut tentang manajemen diri, lihatMengedit Pengaturan Pengguna.

Menyebarkan

Tindakan berikut diizinkan oleh level Deploy, selain tindakan yang diizinkan oleh level Tampilkan.

```
CreateDeployment
UpdateApp
```

Mengelola

Tindakan berikut diizinkan oleh tingkat Kelola, selain tindakan yang diizinkan oleh tingkat Deploy dan Tampilkan.

```
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
```

Izin yang Memerlukan Kebijakan IAM

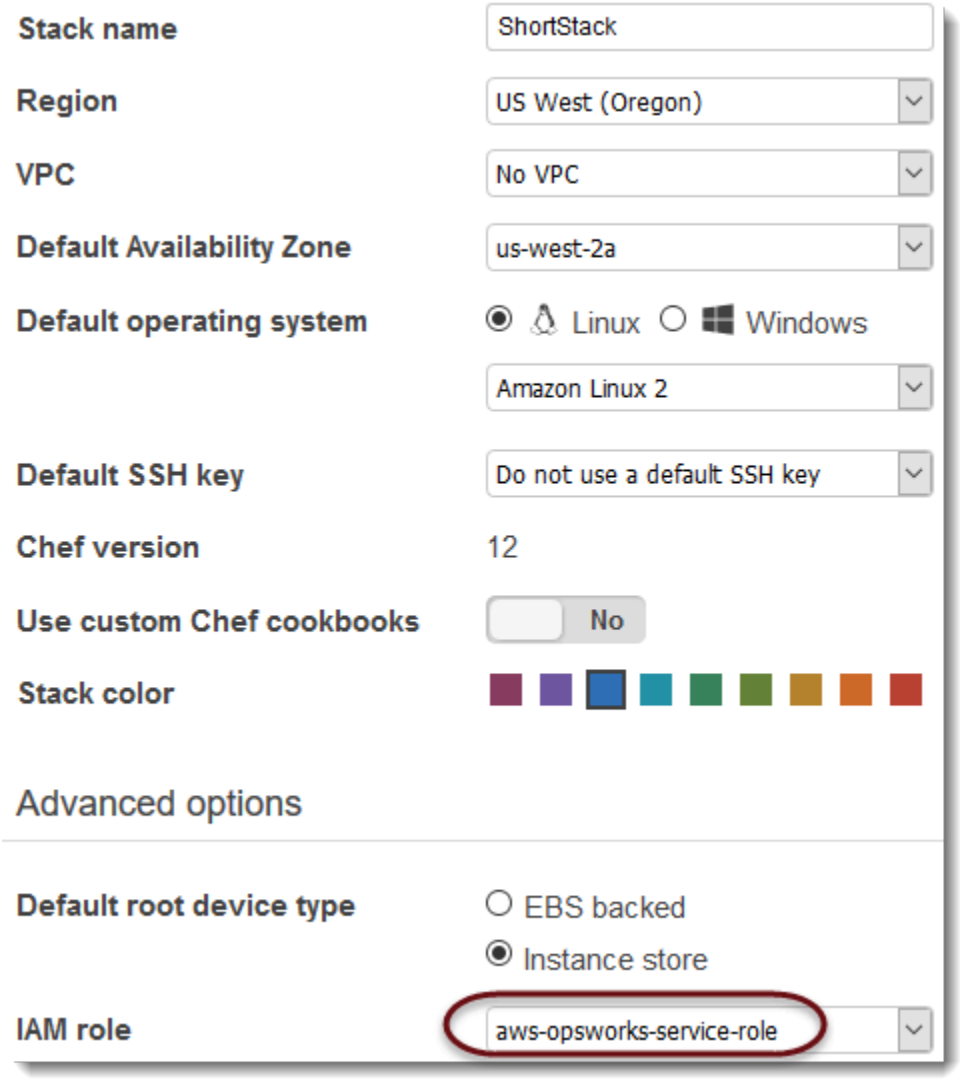
Anda harus memberikan izin untuk tindakan berikut dengan menerapkan kebijakan IAM yang sesuai kepada pengguna. Sebagai contoh, lihat Contoh kebijakan.

```
CloneStack
CreateStack
CreateUserProfile
DeleteUserProfile
DescribeUserProfiles
UpdateUserProfile
```

Mengizinkan AWS OpsWorks Tumpukan untuk Bertindak Atas Nama Anda

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks perlu berinteraksi dengan berbagai layanan AWS atas nama Anda. Misalnya, AWS OpsWorks Stacks berinteraksi dengan Amazon EC2 untuk membuat instance dan dengan Amazon CloudWatch untuk mendapatkan statistik pemantauan. Saat Anda membuat tumpukan, Anda menentukan peran IAM, biasanya disebut peran layanan, yang memberikan AWS OpsWorks Stacks izin yang sesuai.



Saat menentukan peran layanan tumpukan baru, Anda dapat melakukan salah satu hal berikut:

- Tentukan peran layanan standar yang Anda buat sebelumnya.

  Anda biasanya dapat membuat peran layanan standar saat membuat tumpukan pertama, dan kemudian menggunakan peran itu untuk semua tumpukan berikutnya.
- Tentukan peran layanan kustom yang Anda buat menggunakan konsol IAM atau API.

  Pendekatan ini berguna jika Anda ingin memberikan AWS OpsWorks Stacks izin yang lebih terbatas daripada peran layanan standar.

> **Note**
>
> Untuk membuat tumpukan pertama Anda, Anda harus memiliki izin yang ditentukan dalam template AdministratorAccesskebijakan IAM. Izin ini memungkinkan AWS OpsWorks Stacks untuk membuat peran layanan IAM baru dan memungkinkan Anda untuk mengimpor pengguna, seperti yang dijelaskan sebelumnya. Untuk semua tumpukan berikutnya, pengguna dapat memilih peran layanan yang dibuat untuk tumpukan pertama; mereka tidak memerlukan izin administratif penuh untuk membuat tumpukan.

Peran layanan standar memberikan izin berikut:

- Lakukan semua EC2 tindakan Amazon (`ec2:*`).
- Dapatkan CloudWatch statistik (`cloudwatch:GetMetricStatistics`).
- Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas ke server ()`elasticloadbalancing:*`.
- Gunakan instance Amazon RDS sebagai server database (`rds:*`).
- Gunakan peran IAM (`iam:PassRole`) untuk menyediakan komunikasi yang aman antara AWS OpsWorks Stacks dan instans Amazon EC2 Anda.

Jika Anda membuat peran layanan kustom, Anda harus memastikan bahwa itu memberikan semua izin yang dibutuhkan AWS OpsWorks Stacks untuk mengelola tumpukan Anda. Contoh JSON berikut adalah pernyataan kebijakan untuk peran layanan standar; peran layanan kustom harus menyertakan setidaknya izin berikut dalam pernyataan kebijakannya.

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

Peran layanan juga memiliki hubungan kepercayaan. Peran layanan yang dibuat oleh AWS OpsWorks Stacks memiliki hubungan kepercayaan berikut.

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

Peran layanan harus memiliki hubungan kepercayaan ini agar AWS OpsWorks Stacks bertindak atas nama Anda. Jika Anda menggunakan peran layanan default, jangan mengubah hubungan kepercayaan. Jika Anda membuat peran layanan kustom, tentukan hubungan kepercayaan dengan melakukan salah satu hal berikut:

- Jika Anda menggunakan wizard Buat peran di konsol IAM, di Pilih kasus penggunaan, pilih OPSWORKS. Peran ini memiliki hubungan kepercayaan yang sesuai, tetapi tidak ada kebijakan yang dilampirkan secara implisit. Untuk memberikan izin AWS OpsWorks Stacks untuk bertindak atas nama Anda, buat kebijakan yang dikelola pelanggan yang berisi hal-hal berikut, dan lampirkan ke peran baru.

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

- Jika Anda menggunakan AWS CloudFormation template, Anda dapat menambahkan sesuatu seperti berikut ke bagian Sumber Daya template Anda.

```
"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": {
              ...
            } ]
        }
      },
   }
}
```

Pencegahan deputi kebingungan lintas layanan di Stacks AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Masalah "confused deputy" adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang memilik hak akses lebih tinggi untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses.

Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan dalam kebijakan akses tumpukan untuk membatasi izin yang diberikan AWS OpsWorks Stacks kepada layanan lain ke tumpukan. Jika nilai `aws:SourceArn` tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global tersebut untuk membatasi izin. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama. Gunakan `aws:SourceArn` jika Anda ingin hanya satu tumpukan dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan tumpukan apa pun di akun itu dikaitkan dengan penggunaan lintas layanan.

Nilai `aws:SourceArn` harus ARN dari tumpukan. AWS OpsWorks

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci konteks kondisi `aws:SourceArn` global dengan ARN penuh dari AWS OpsWorks tumpukan Stacks. Jika Anda tidak mengetahui ARN lengkap, atau jika Anda menentukan beberapa tumpukan ARNs, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Bagian berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan di AWS OpsWorks Tumpukan untuk mencegah masalah wakil yang membingungkan.

Mencegah eksploitasi wakil yang membingungkan di Stacks AWS OpsWorks

Bagian ini menjelaskan bagaimana Anda dapat membantu mencegah eksploitasi deputi yang membingungkan di AWS OpsWorks Tumpukan, dan menyertakan contoh kebijakan izin yang dapat Anda lampirkan ke peran IAM yang Anda gunakan untuk mengakses Tumpukan. AWS OpsWorks Sebagai praktik terbaik keamanan, kami sarankan untuk menambahkan kunci `aws:SourceArn` dan `aws:SourceAccount` kondisi ke hubungan kepercayaan yang dimiliki peran IAM Anda dengan layanan lain. Hubungan kepercayaan memungkinkan AWS OpsWorks Stacks untuk mengambil peran untuk melakukan tindakan di layanan lain yang diperlukan untuk membuat atau mengelola AWS OpsWorks tumpukan Stacks Anda.

Untuk mengedit hubungan kepercayaan untuk menambahkan **`aws:SourceArn`** dan **`aws:SourceAccount`** mengkondisikan kunci

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Peran.
3. Di kotak Pencarian, cari peran yang Anda gunakan untuk akses ke AWS OpsWorks Tumpukan. Peran yang AWS dikelola adalah`aws-opsworks-service-role`.
4. Pada halaman Ringkasan untuk peran tersebut, pilih tab Trust relationship.
5. Pada tab Hubungan kepercayaan, pilih Edit kebijakan kepercayaan.
6. Pada halaman Edit kebijakan kepercayaan, tambahkan setidaknya satu dari `aws:SourceArn` atau kunci `aws:SourceAccount` kondisi ke kebijakan. Gunakan `aws:SourceArn` untuk membatasi hubungan kepercayaan antara layanan lintas (seperti Amazon EC2) dan AWS OpsWorks Tumpukan ke tumpukan AWS OpsWorks Stacks tertentu, yang lebih ketat. Tambahkan `aws:SourceAccount` untuk membatasi hubungan kepercayaan antara layanan lintas dan AWS OpsWorks Tumpukan ke tumpukan di akun tertentu, yang kurang membatasi. Berikut adalah contohnya. Perhatikan bahwa jika Anda menggunakan kedua tombol kondisi, akun IDs harus sama.

```
{
  "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. Setelah selesai menambahkan tombol kondisi, pilih Perbarui kebijakan.

Berikut ini adalah contoh tambahan peran yang membatasi akses ke tumpukan dengan menggunakan `aws:SourceArn` dan`aws:SourceAccount`.

Topik

- Contoh: Mengakses tumpukan di wilayah tertentu
- Contoh: Menambahkan lebih dari satu tumpukan ARN ke aws:SourceArn

Contoh: Mengakses tumpukan di wilayah tertentu

Pernyataan hubungan kepercayaan peran berikut mengakses AWS OpsWorks tumpukan Stacks di Wilayah Timur AS (Ohio) (). `us-east-2` Perhatikan bahwa wilayah ditentukan dalam nilai ARN`aws:SourceArn`, tetapi nilai ID tumpukan adalah wildcard (*).

```
{
  "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/*"
        }
      }
    }
  ]
}
```

Contoh: Menambahkan lebih dari satu tumpukan ARN ke **`aws:SourceArn`**

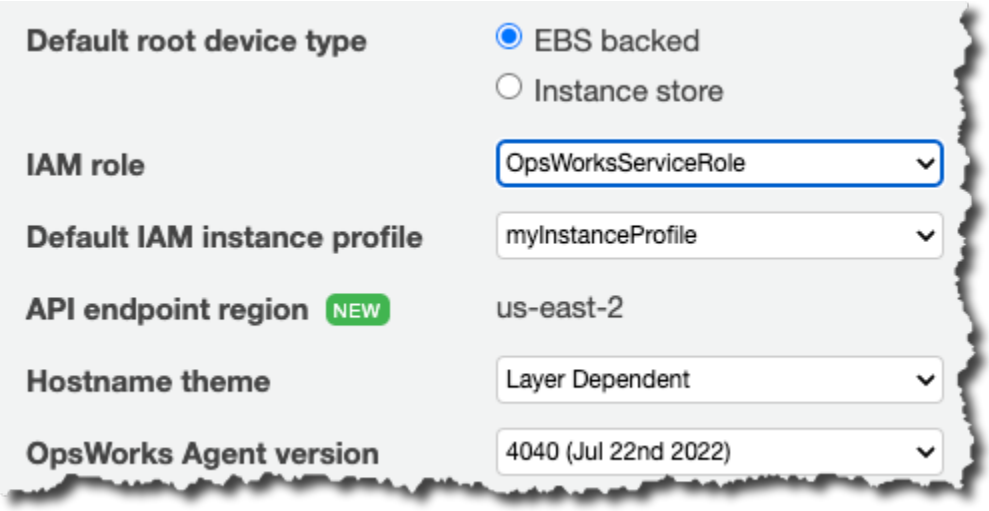
Contoh berikut membatasi akses ke array dua AWS OpsWorks tumpukan Stacks di ID akun 123456789012.

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

Menentukan Izin untuk Aplikasi yang Berjalan pada instance EC2

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika aplikasi yang berjalan di EC2 instans Amazon stack Anda perlu mengakses sumber daya AWS lainnya, seperti bucket Amazon S3, aplikasi tersebut harus memiliki izin yang sesuai. Untuk memberikan izin tersebut, Anda menggunakan profil instance. Anda dapat menentukan profil instance untuk setiap instance saat membuat tumpukan AWS OpsWorks Stacks.



Anda juga dapat menentukan profil untuk instance lapisan dengan mengedit konfigurasi lapisan.

Profil instance menentukan peran IAM. Aplikasi yang berjalan pada instans dapat mengasumsikan peran tersebut untuk mengakses sumber daya AWS, tunduk pada izin yang diberikan oleh kebijakan peran tersebut. Untuk informasi selengkapnya tentang cara aplikasi mengasumsikan peran, lihat Mengasumsikan Peran Menggunakan Panggilan API.

Anda dapat membuat profil instance dengan salah satu cara berikut:

- Gunakan konsol IAM atau API untuk membuat profil.

  Untuk informasi selengkapnya, lihat Peran (Delegasi dan Federasi).
- Gunakan AWS CloudFormation template untuk membuat profil.

  Untuk beberapa contoh cara menyertakan sumber daya IAM dalam template, lihat Identity and Access Management (IAM) Template Snippets.

Profil instans harus memiliki hubungan kepercayaan dan kebijakan terlampir yang memberikan izin untuk mengakses sumber daya AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Profil instans harus memiliki hubungan kepercayaan ini agar AWS OpsWorks Stacks bertindak atas nama Anda. Jika Anda menggunakan peran layanan default, jangan mengubah hubungan kepercayaan. Jika Anda membuat peran layanan khusus, tentukan kapal hubungan kepercayaan sebagai berikut:

- Jika Anda menggunakan wizard Buat Peran di konsol IAM, tentukan jenis EC2 peran Amazon di bawah AWS Service Roles di halaman kedua wizard.
- Jika Anda menggunakan AWS CloudFormation template, Anda dapat menambahkan sesuatu seperti berikut ini ke bagian Sumber Daya template Anda.

```
"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"
            }
         ]
      }
```

```
    }
}
```

Saat membuat profil instans, Anda dapat melampirkan kebijakan yang sesuai ke peran profil pada saat itu. Setelah membuat tumpukan, Anda harus menggunakan konsol IAM atau API untuk melampirkan kebijakan yang sesuai ke peran profil. Misalnya, kebijakan berikut memberikan akses penuh ke semua objek di bucket Amazon S3 bernama amzn-s3-demo-bucket. Ganti *region* dan amzn-s3-demo-bucket dengan nilai yang sesuai dengan konfigurasi Anda.

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

Untuk contoh cara membuat dan menggunakan profil instans, lihat Menggunakan Bucket Amazon S3.

Jika aplikasi Anda menggunakan profil instans untuk memanggil AWS OpsWorks Stacks API dari EC2 instans, kebijakan tersebut harus mengizinkan `iam:PassRole` tindakan tersebut selain tindakan yang sesuai untuk AWS OpsWorks Stacks dan layanan AWS lainnya. `iam:PassRole`Izin memungkinkan AWS OpsWorks Stacks untuk mengambil peran layanan atas nama Anda. Untuk informasi selengkapnya tentang AWS OpsWorks Stacks API, lihat AWS OpsWorks API Referensi.

Berikut ini adalah contoh kebijakan IAM yang memungkinkan Anda memanggil tindakan AWS OpsWorks Stacks apa pun dari sebuah EC2 instance, serta tindakan Amazon EC2 atau Amazon S3 apa pun.

```
{
  "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**
>
> Jika Anda tidak mengizinkan`iam:PassRole`, upaya apa pun untuk memanggil tindakan AWS OpsWorks Stacks gagal dengan kesalahan seperti berikut:
>
> ```
> User: arn:aws:sts::123456789012:federated-user/Bob is not authorized
> to perform: iam:PassRole on resource:
> arn:aws:sts::123456789012:role/OpsWorksStackIamRole
> ```

Untuk informasi selengkapnya tentang penggunaan peran pada EC2 instans untuk izin, lihat Memberikan Aplikasi yang Berjalan di EC2 Instans Amazon Akses ke Sumber Daya AWS di Panduan Pengguna.AWS Identity and Access Management

Mengelola Akses SSH

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks mendukung kunci SSH untuk tumpukan Linux dan Windows.

- Untuk instance Linux, Anda dapat menggunakan SSH untuk masuk ke sebuah instance, misalnya, untuk menjalankan perintah CLI agen.

  Untuk informasi selengkapnya, lihat Login dengan SSH.
- Untuk instance Windows, Anda dapat menggunakan kunci SSH untuk mendapatkan kata sandi Administrator instans, yang kemudian dapat Anda gunakan untuk masuk dengan RDP.

  Untuk informasi selengkapnya, lihat Masuk dengan RDP.

Otentikasi didasarkan pada key pair SSH, yang terdiri dari public key dan private key:

- Anda menginstal kunci publik pada instance.

  Lokasi tergantung pada sistem operasi tertentu, tetapi AWS OpsWorks Stacks menangani detail untuk Anda.
- Anda menyimpan kunci pribadi secara lokal dan menyediakannya ke klien SSH, seperti `ssh.exe`, untuk mengakses instance.

  Klien SSH menggunakan kunci pribadi untuk terhubung ke instance.

Untuk menyediakan akses SSH ke pengguna stack, Anda memerlukan cara untuk membuat pasangan kunci SSH, menginstal kunci publik pada instance tumpukan, dan mengelola kunci pribadi dengan aman.

Amazon EC2 menyediakan cara sederhana untuk menginstal kunci SSH publik pada sebuah instance. Anda dapat menggunakan EC2 konsol Amazon atau API untuk membuat satu atau beberapa pasangan kunci untuk setiap wilayah AWS yang ingin Anda gunakan. Amazon EC2 menyimpan kunci publik di AWS dan Anda menyimpan kunci pribadi secara lokal. Saat meluncurkan instance, Anda menentukan salah satu pasangan kunci wilayah tersebut dan Amazon EC2 secara otomatis menginstalnya pada instance. Anda kemudian menggunakan kunci pribadi yang sesuai untuk masuk ke instance. Untuk informasi selengkapnya, lihat Pasangan EC2 Kunci Amazon.

Dengan AWS OpsWorks Stacks, Anda dapat menentukan salah satu pasangan EC2 kunci Amazon di wilayah tersebut saat membuat tumpukan, dan secara opsional menggantinya dengan key pair yang berbeda saat membuat setiap instance. Ketika AWS OpsWorks Stacks meluncurkan EC2 instans Amazon yang sesuai, Stacks menentukan key pair dan Amazon EC2 menginstal kunci publik pada instance. Anda kemudian dapat menggunakan kunci pribadi untuk masuk atau mengambil

kata sandi Administrator, seperti yang Anda lakukan dengan EC2 instance Amazon standar. Untuk informasi selengkapnya, lihat Memasang EC2 Kunci Amazon.

Menggunakan EC2 key pair Amazon nyaman, tetapi memiliki dua keterbatasan signifikan:

- Amazon EC2 key pair terkait dengan wilayah AWS tertentu.

  Jika Anda bekerja di beberapa wilayah, Anda harus mengelola beberapa pasangan kunci.
- Anda hanya dapat menginstal satu Amazon EC2 key pair pada sebuah instance.

  Jika Anda ingin mengizinkan beberapa pengguna untuk masuk, mereka semua harus memiliki salinan kunci pribadi, yang bukan merupakan praktik keamanan yang disarankan.

Untuk tumpukan Linux, AWS OpsWorks Stacks menyediakan cara yang lebih sederhana dan lebih fleksibel untuk mengelola pasangan kunci SSH.

- Setiap pengguna mendaftarkan personal key pair.

  Mereka menyimpan kunci pribadi secara lokal dan mendaftarkan kunci publik dengan AWS OpsWorks Stacks, seperti yang dijelaskan dalam. Mendaftarkan Kunci SSH Publik Pengguna
- Saat Anda menetapkan izin pengguna untuk tumpukan, Anda menentukan pengguna mana yang harus memiliki akses SSH ke instance tumpukan.

  AWS OpsWorks Stacks secara otomatis membuat pengguna sistem pada instance stack untuk setiap pengguna yang berwenang dan menginstal kunci publik mereka. Pengguna kemudian dapat menggunakan kunci pribadi yang sesuai untuk masuk, seperti yang dijelaskan dalamLogin dengan SSH.

Menggunakan kunci SSH pribadi memiliki keuntungan sebagai berikut.

- Tidak perlu mengonfigurasi kunci secara manual pada instance; AWS OpsWorks Stacks secara otomatis menginstal kunci publik yang sesuai pada setiap instance.
- AWS OpsWorks Stacks hanya menginstal kunci publik pribadi pengguna resmi.

  Pengguna yang tidak sah tidak dapat menggunakan kunci pribadi pribadi mereka untuk mendapatkan akses ke instans. Dengan pasangan EC2 kunci Amazon, setiap pengguna dengan kunci pribadi yang sesuai dapat masuk, dengan atau tanpa akses SSH resmi.

- Jika pengguna tidak lagi membutuhkan akses SSH, Anda dapat menggunakan halaman Izin untuk mencabut izin SSH/RDP pengguna.

  AWS OpsWorks Tumpukan segera menghapus kunci publik dari instance tumpukan.
- Anda dapat menggunakan kunci yang sama untuk wilayah AWS mana pun.

  Pengguna harus mengelola hanya satu kunci pribadi.
- Tidak perlu berbagi kunci pribadi.

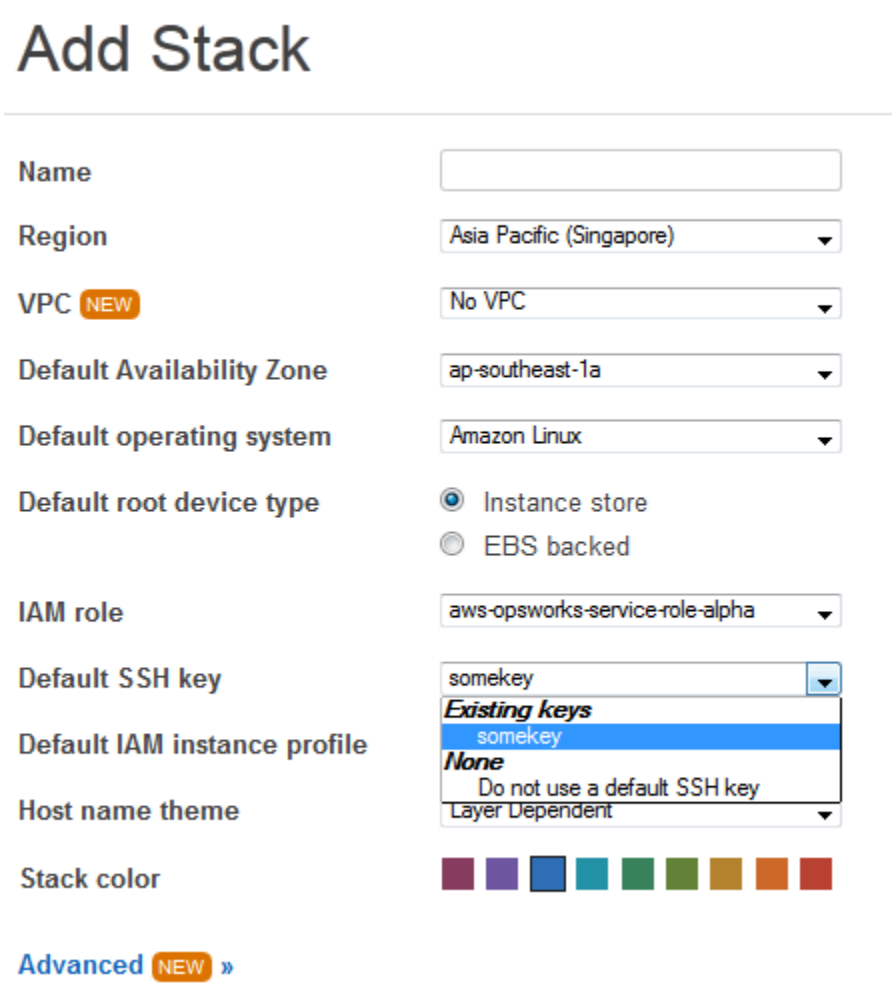
  Setiap pengguna memiliki kunci pribadinya sendiri.
- Sangat mudah untuk memutar tombol.

  Anda atau pengguna memperbarui kunci publik di Pengaturan Saya dan AWS OpsWorks Tumpukan secara otomatis memperbarui instance.

Memasang EC2 Kunci Amazon

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

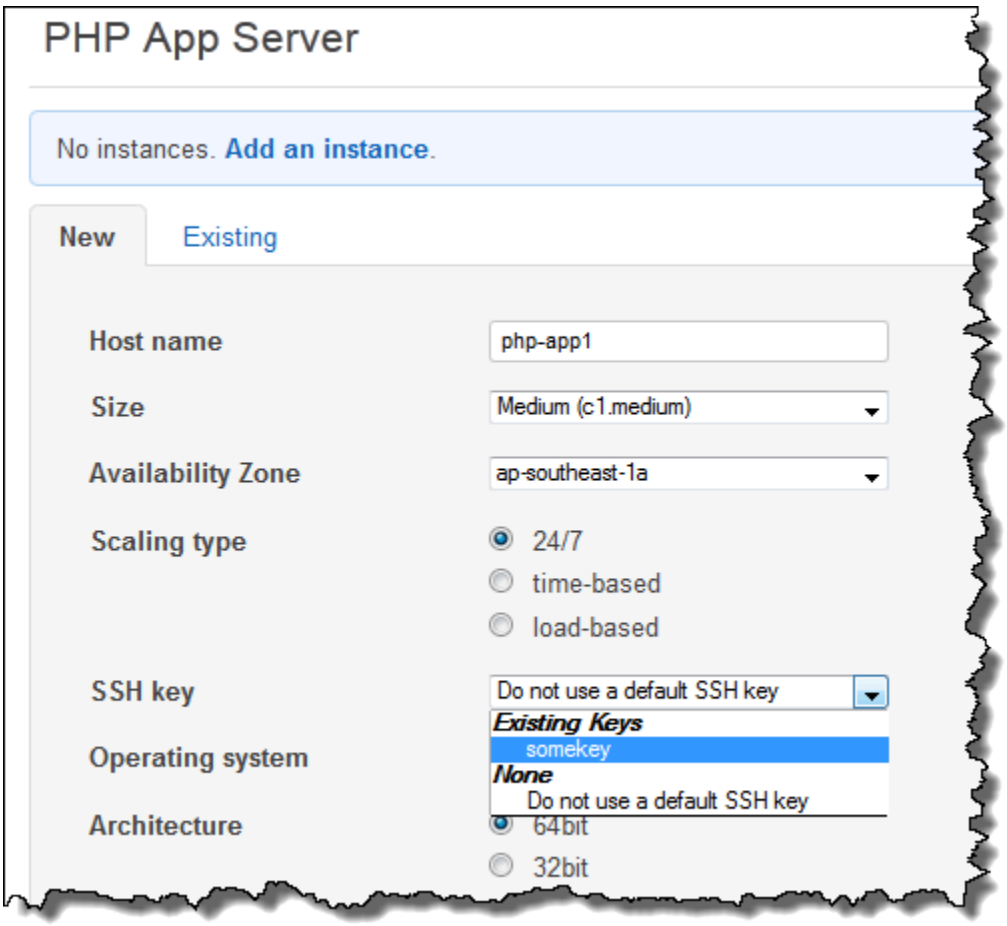
Saat membuat tumpukan, Anda dapat menentukan kunci EC2 SSH Amazon yang diinstal secara default pada setiap instance di tumpukan.



Daftar kunci SSH default menunjukkan Amazon EC2keys akun AWS Anda. Anda dapat melakukan salah satu dari yang berikut:

- Pilih kunci yang sesuai dari daftar.
- Pilih Jangan gunakan kunci SSH default untuk menentukan tidak ada kunci.

Jika Anda memilih Jangan gunakan kunci SSH default, atau Anda ingin mengganti kunci default tumpukan, Anda dapat menentukan kunci saat membuat instance.



Ketika Anda memulai instance AWS OpsWorks Stacks menginstal kunci publik dalam file. `authorized_keys`

Mendaftarkan Kunci SSH Publik Pengguna

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Ada dua cara untuk mendaftarkan kunci SSH publik pengguna:

- Pengguna administratif dapat menetapkan kunci SSH publik untuk satu atau lebih pengguna dan memberi mereka kunci pribadi yang sesuai.
- Pengguna administratif dapat mengaktifkan manajemen mandiri untuk satu atau lebih pengguna.

  Pengguna tersebut kemudian dapat menentukan kunci SSH publik mereka sendiri.

Untuk informasi selengkapnya bagaimana pengguna administratif dapat mengaktifkan manajemen mandiri atau menetapkan kunci publik kepada pengguna, lihatMengedit Pengaturan Pengguna.

Menghubungkan ke instance berbasis Linux dengan menggunakan SSH di terminal PuTTY memerlukan langkah tambahan. Untuk informasi selengkapnya, lihat Menghubungkan ke Instans Linux Anda dari Windows Menggunakan PuTTY dan Pemecahan Masalah Menyambung ke Instans Anda di dokumentasi AWS.

Berikut ini menjelaskan bagaimana pengguna dengan manajemen mandiri diaktifkan dapat menentukan kunci publik mereka.

Untuk menentukan kunci publik SSH

1. Buat key pair SSH.

   Pendekatan paling sederhana adalah menghasilkan key pair secara lokal. Untuk informasi selengkapnya, lihat Cara Menghasilkan Kunci Anda Sendiri dan Mengimpornya ke Amazon EC2.

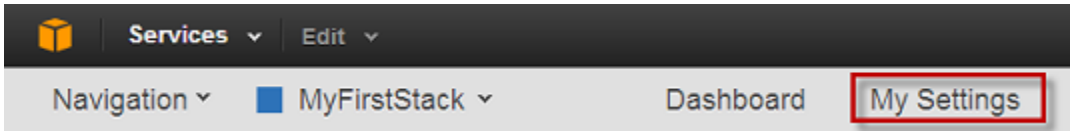
   > **Note**
   >
   > Jika Anda menggunakan Pu TTYgen untuk menghasilkan key pair Anda, salin kunci publik dari kunci Publik untuk ditempelkan ke kotak file OpenSSH authorized_keys. Mengklik Simpan Kunci Publik menyimpan kunci publik dalam format yang tidak didukung oleh MindTerm.

2. Masuk ke konsol AWS OpsWorks Stacks sebagai pengguna IAM dengan pengelolaan mandiri diaktifkan.

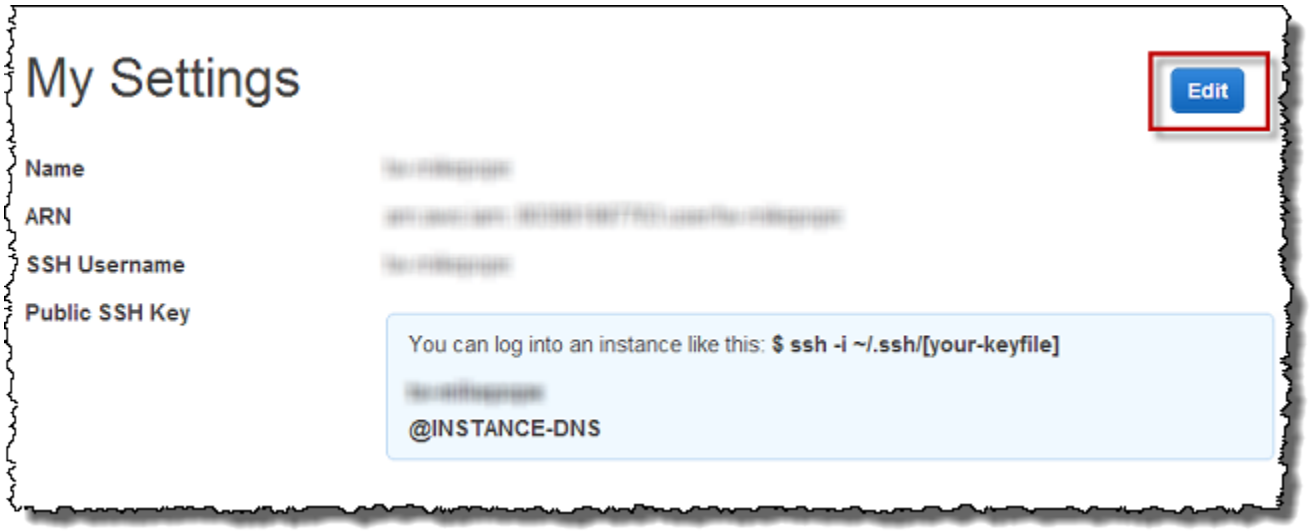
   > **Important**
   >
   > Jika Anda masuk sebagai pemilik akun, atau sebagai pengguna IAM yang tidak mengaktifkan pengelolaan mandiri, AWS OpsWorks Stacks tidak menampilkan Pengaturan Saya. Jika Anda adalah pengguna administratif atau pemilik akun, Anda

dapat menentukan kunci SSH dengan membuka halaman Pengguna dan mengedit pengaturan pengguna.

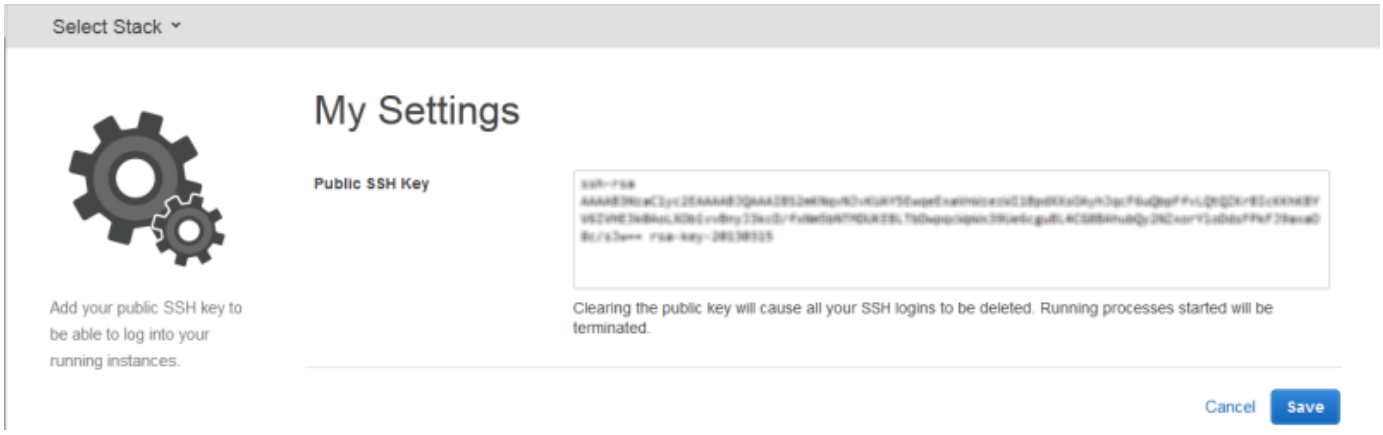
3. Pilih Pengaturan Saya, yang menampilkan pengaturan untuk pengguna yang masuk.



4. Pada halaman Pengaturan Saya, klik Edit.



5. Di kotak Kunci SSH Publik, masukkan kunci publik SSH Anda, lalu klik Simpan.



Important

Untuk menggunakan klien MindTerm SSH bawaan untuk terhubung ke EC2 instans Amazon, pengguna harus masuk sebagai pengguna IAM dan memiliki kunci SSH publik yang terdaftar

di Stacks. AWS OpsWorks Untuk informasi selengkapnya, lihat Menggunakan Built-in MindTerm SSH Client.

Mengelola Pembaruan Keamanan Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pembaruan Keamanan

Penyedia sistem operasi Linux menyediakan pembaruan rutin, yang sebagian besar adalah patch keamanan sistem operasi tetapi juga dapat menyertakan pembaruan untuk paket yang diinstal. Anda harus memastikan bahwa sistem operasi instans Anda terkini dengan patch keamanan terbaru.

Secara default, AWS OpsWorks Stacks secara otomatis menginstal pembaruan terbaru selama penyiapan, setelah instance selesai booting. AWS OpsWorks Stacks tidak secara otomatis menginstal pembaruan setelah instance online, untuk menghindari gangguan seperti memulai ulang server aplikasi. Sebagai gantinya, Anda mengelola pembaruan untuk instans online Anda sendiri, sehingga Anda dapat meminimalkan gangguan apa pun.

Kami menyarankan Anda menggunakan salah satu dari berikut ini untuk memperbarui instans online Anda.

- Buat dan mulai instance baru untuk menggantikan instans online Anda saat ini. Kemudian hapus instance saat ini.

  Instans baru akan memiliki set patch keamanan terbaru yang diinstal selama pengaturan.
- Pada instance berbasis Linux di Chef 11.10 atau tumpukan yang lebih lama, jalankan perintah tumpukan Update Dependencies, yang menginstal kumpulan patch keamanan saat ini dan pembaruan lainnya pada instance yang ditentukan.

Untuk kedua pendekatan ini, AWS OpsWorks Stacks melakukan pembaruan dengan menjalankan `yum update` Amazon Linux dan Red Hat Enterprise Linux (RHEL) atau `apt-get update` untuk Ubuntu. Setiap distribusi menangani pembaruan agak berbeda, jadi Anda harus memeriksa informasi di tautan terkait untuk memahami dengan tepat bagaimana pembaruan akan memengaruhi instance Anda:

- Amazon Linux — Pembaruan Amazon Linux menginstal patch keamanan dan mungkin juga menginstal pembaruan fitur, termasuk pembaruan paket.

  Untuk informasi lebih lanjut, lihat Amazon Linux AMI FAQs.
- Ubuntu — Pembaruan Ubuntu sebagian besar terbatas untuk menginstal patch keamanan, tetapi mungkin juga menginstal pembaruan paket untuk sejumlah perbaikan kritis yang terbatas.

  Untuk informasi selengkapnya, lihat LTS - Ubuntu Wiki.
- CentOS — Pembaruan CentOS umumnya mempertahankan kompatibilitas biner dengan versi sebelumnya.
- RHEL — Pembaruan RHEL umumnya mempertahankan kompatibilitas biner dengan versi sebelumnya.

  Untuk informasi selengkapnya, lihat Red Hat Enterprise Linux Life Cycle.

Jika Anda menginginkan kontrol lebih besar atas pembaruan, seperti menentukan versi paket tertentu, Anda dapat menonaktifkan pembaruan otomatis dengan menggunakan,, CreateInstanceUpdateInstanceCreateLayer, atau UpdateLayertindakan—atau metode AWS SDK atau perintah AWS CLI—untuk menyetel parameternya. `InstallUpdatesOnBoot` `false` Contoh berikut menunjukkan cara menggunakan AWS CLI untuk menonaktifkan `InstallUpdatesOnBoot` sebagai pengaturan default untuk lapisan yang ada.

```
aws opsworks update-layer --layer-id layer ID --no-install-updates-on-boot
```

Anda kemudian harus mengelola pembaruan sendiri. Misalnya, Anda dapat menggunakan salah satu strategi ini:

- Terapkan resep khusus yang menjalankan perintah shell yang sesuai untuk menginstal pembaruan pilihan Anda.

  Karena pembaruan sistem tidak memetakan secara alami ke acara siklus hidup, sertakan resep di buku masak khusus Anda tetapi jalankan secara manual. Untuk pembaruan paket, Anda juga

dapat menggunakan sumber daya yum_package (Amazon Linux) atau apt_package (Ubuntu) alih-alih perintah shell.

- Masuk ke setiap instance dengan SSH dan jalankan perintah yang sesuai secara manual.

Menggunakan Grup Keamanan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Grup Keamanan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setiap EC2 instans Amazon memiliki satu atau lebih grup keamanan terkait yang mengatur lalu lintas jaringan instans, seperti firewall. Grup keamanan memiliki satu atau lebih aturan, yang masing-masing menentukan kategori tertentu dari lalu lintas yang diizinkan. Aturan menentukan hal berikut:

- Jenis lalu lintas yang diizinkan, seperti SSH atau HTTP
- Protokol lalu lintas, seperti TCP atau UDP
- Rentang alamat IP tempat lalu lintas dapat berasal
- Jangkauan port yang diizinkan lalu lintas

Kelompok keamanan memiliki dua jenis aturan:

- Aturan masuk mengatur lalu lintas jaringan masuk.

  Misalnya, instance server aplikasi biasanya memiliki aturan inbound yang memungkinkan lalu lintas HTTP masuk dari alamat IP apa pun ke port 80, dan aturan masuk lain yang memungkinkan lalu lintas SSH masuk ke port 22 dari kumpulan alamat IP yang ditentukan.
- Aturan keluar mengatur lalu lintas jaringan keluar.

  Praktik umum adalah menggunakan pengaturan default, yang memungkinkan lalu lintas keluar.

Untuk informasi selengkapnya tentang grup keamanan, lihat Grup EC2 Keamanan Amazon.

Pertama kali Anda membuat tumpukan di suatu wilayah, AWS OpsWorks Stacks membuat grup keamanan bawaan untuk setiap lapisan dengan seperangkat aturan yang sesuai. Semua grup memiliki aturan keluar default, yang memungkinkan semua lalu lintas keluar. Secara umum, aturan masuk memungkinkan hal berikut:

- Lalu lintas TCP, UDP, dan ICMP masuk dari lapisan Stacks yang sesuai AWS OpsWorks
- Lalu lintas TCP masuk pada port 22 (login SSH)

> **Warning**
>
> Konfigurasi grup keamanan default membuka SSH (port 22) ke lokasi jaringan apa pun (0.0.0.0/0.) Hal ini memungkinkan semua alamat IP untuk mengakses instans Anda dengan menggunakan SSH. Untuk lingkungan produksi, Anda harus menggunakan konfigurasi yang hanya memungkinkan akses SSH dari alamat IP tertentu atau rentang alamat. Perbarui grup keamanan default segera setelah dibuat, atau gunakan grup keamanan khusus sebagai gantinya.

- Untuk lapisan server web, semua lalu lintas TCP masuk, dan UDP ke port 80 (HTTP) dan 443 (HTTPS)

> **Note**
>
> Grup `AWS-OpsWorks-RDP-Server` keamanan bawaan ditugaskan ke semua instance Windows untuk memungkinkan akses RDP. Namun, secara default, itu tidak memiliki aturan apa pun. Jika Anda menjalankan tumpukan Windows dan ingin menggunakan RDP untuk

mengakses instance, Anda harus menambahkan aturan masuk yang memungkinkan akses RDP. Untuk informasi selengkapnya, lihat Masuk dengan RDP.

Untuk melihat detail untuk setiap grup, buka EC2konsol Amazon, pilih Grup Keamanan di panel navigasi, dan pilih grup keamanan lapisan yang sesuai. Misalnya, AWS- OpsWorks -Default-Server adalah grup keamanan bawaan default untuk semua tumpukan, dan AWS- OpsWorks - WebApp adalah grup keamanan bawaan default untuk tumpukan sampel Chef 12.

Note

Jika Anda secara tidak sengaja menghapus grup keamanan AWS OpsWorks Stacks, cara yang lebih disukai untuk membuatnya adalah dengan meminta AWS OpsWorks Stacks melakukan tugas untuk Anda. Cukup buat tumpukan baru di wilayah AWS yang sama—dan VPC, jika AWS OpsWorks ada—dan Stacks akan secara otomatis membuat ulang semua grup keamanan bawaan, termasuk yang Anda hapus. Anda kemudian dapat menghapus tumpukan jika Anda tidak memiliki penggunaan lebih lanjut untuk itu; grup keamanan akan tetap ada. Jika Anda ingin membuat ulang grup keamanan secara manual, itu harus berupa duplikat persis dari aslinya, termasuk kapitalisasi nama grup.
Selain itu, AWS OpsWorks Stacks akan mencoba membuat ulang semua grup keamanan bawaan jika salah satu hal berikut terjadi:

- Anda membuat perubahan apa pun pada halaman pengaturan tumpukan di konsol AWS OpsWorks Stacks.
- Anda memulai salah satu contoh tumpukan.
- Anda membuat tumpukan baru.

Anda dapat menggunakan salah satu dari pendekatan berikut untuk menentukan kelompok keamanan. Anda menggunakan pengaturan Gunakan grup OpsWorks keamanan untuk menentukan preferensi Anda saat membuat tumpukan.

- Ya (pengaturan default) - AWS OpsWorks Tumpukan secara otomatis mengaitkan grup keamanan bawaan yang sesuai dengan setiap lapisan.

  Anda dapat menyempurnakan grup keamanan bawaan lapisan dengan menambahkan grup keamanan khusus dengan pengaturan pilihan Anda. Namun, ketika Amazon EC2 mengevaluasi beberapa grup keamanan, Amazon menggunakan aturan yang paling tidak membatasi, sehingga

Anda tidak dapat menggunakan pendekatan ini untuk menentukan aturan yang lebih ketat daripada grup bawaan.

- Tidak - AWS OpsWorks Tumpukan tidak mengaitkan grup keamanan bawaan dengan lapisan.

  Anda harus membuat grup keamanan yang sesuai dan mengaitkan setidaknya satu dengan setiap lapisan yang Anda buat. Gunakan pendekatan ini untuk menentukan aturan yang lebih ketat daripada grup bawaan. Perhatikan bahwa Anda masih dapat secara manual mengaitkan grup keamanan bawaan dengan lapisan jika Anda mau; grup keamanan khusus hanya diperlukan untuk lapisan yang memerlukan pengaturan khusus.

> **Important**
>
> Jika Anda menggunakan grup keamanan bawaan, Anda tidak dapat membuat aturan yang lebih ketat dengan memodifikasi pengaturan grup secara manual. Setiap kali Anda membuat tumpukan, AWS OpsWorks Stacks menimpa konfigurasi grup keamanan bawaan, sehingga setiap perubahan yang Anda buat akan hilang saat berikutnya Anda membuat tumpukan. Jika lapisan memerlukan pengaturan grup keamanan yang lebih ketat daripada grup keamanan bawaan, atur Gunakan grup OpsWorks keamanan ke Tidak, buat grup keamanan khusus dengan pengaturan pilihan Anda, dan tetapkan ke lapisan pada pembuatan.

AWS OpsWorks Dukungan Stacks untuk Chef 12 Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini memberikan gambaran singkat tentang AWS OpsWorks Stacks for Chef 12 Linux. Untuk informasi tentang Chef 12 di Windows, lihatMemulai: Windows. Untuk informasi tentang versi Chef sebelumnya di Linux, lihatChef 11.10 dan Versi Sebelumnya untuk Linux.

Gambaran Umum

AWS OpsWorks Stacks mendukung Chef 12, Chef versi terbaru, untuk tumpukan Linux. Untuk informasi selengkapnya, lihat Pelajari Chef.

AWS OpsWorks Stacks terus mendukung Chef 11.10 untuk tumpukan Linux. Namun, jika Anda adalah pengguna Chef tingkat lanjut yang ingin mendapatkan manfaat dari banyak pilihan buku masak komunitas atau menulis buku masak khusus Anda sendiri, kami sarankan Anda menggunakan Chef 12. Tumpukan Chef 12 memberikan keunggulan berikut dibandingkan Chef 11.10 dan tumpukan sebelumnya untuk Linux:

- Dua Chef yang terpisah berjalan - Ketika sebuah perintah dijalankan pada sebuah instance, agen AWS OpsWorks Stacks sekarang menjalankan dua proses Chef yang terisolasi: satu run untuk tugas yang mengintegrasikan instance dengan layanan AWS lain seperti AWS Identity and Access Management (IAM), dan satu dijalankan untuk buku masak kustom Anda. Chef run pertama menginstal agen AWS OpsWorks Stacks pada instance dan melakukan tugas sistem seperti pengaturan dan manajemen pengguna, pengaturan dan konfigurasi volume, konfigurasi CloudWatch metrik, dan sebagainya. Run kedua didedikasikan khusus untuk menjalankan resep khusus AndaAWS OpsWorks Acara Siklus Hidup Stacks. Lari kedua ini memungkinkan Anda menggunakan buku masak Chef atau buku masak komunitas Anda sendiri.
- Resolusi konflik namespace - Sebelum Chef 12, AWS OpsWorks Stacks melakukan tugas sistem dan menjalankan resep bawaan dan khusus di lingkungan bersama. Hal ini mengakibatkan konflik namespace dan kurangnya kejelasan tentang resep mana yang telah dijalankan AWS OpsWorks Stacks. Konfigurasi default yang tidak diinginkan harus ditimpa secara manual, tugas yang memakan waktu dan rawan kesalahan. Di Chef 12 untuk Linux, AWS OpsWorks Stacks tidak lagi mendukung buku masak Chef bawaan untuk lingkungan server aplikasi seperti PHP, Node.js, atau Rails. Dengan menghilangkan resep bawaan, AWS OpsWorks Stacks menghilangkan masalah tabrakan penamaan antara resep bawaan dan khusus.
- Dukungan kuat untuk buku masak komunitas Chef — AWS OpsWorks Stacks Chef 12 Linux menawarkan kompatibilitas dan dukungan yang lebih besar untuk buku masak komunitas dari supermarket Chef. Anda sekarang dapat menggunakan buku masak komunitas yang lebih unggul dari buku masak bawaan yang disediakan AWS OpsWorks Stacks sebelumnya — buku masak yang dirancang untuk digunakan dengan lingkungan dan kerangka kerja server aplikasi terbaru. Anda dapat menjalankan sebagian besar buku masak ini tanpa modifikasi pada Chef 12 untuk Linux. Untuk informasi lebih lanjut, kunjungi Chef Supermarket di situs web Learn Chef, situs web Chef Supermarket, dan repositori Chef Cookbooks di. GitHub

- Pembaruan Timely Chef 12 - AWS OpsWorks Stacks akan memperbarui lingkungan Chef-nya ke versi Chef 12 terbaru segera setelah setiap rilis Chef. Dengan Chef 12, pembaruan Chef kecil dan rilis agen AWS OpsWorks Stacks baru akan bertepatan. Ini memungkinkan Anda menguji rilis Chef baru secara langsung, dan memungkinkan resep dan aplikasi Chef Anda memanfaatkan fitur Chef terbaru.

Untuk informasi lebih lanjut tentang versi Chef yang didukung sebelum Chef 12, lihatChef 11.10 dan Versi Sebelumnya untuk Linux.

Pindah ke Chef 12

Perubahan Key AWS OpsWorks Stacks untuk Chef 12 Linux, dibandingkan dengan dukungan untuk Chef versi 11.10, 11.4, dan 0.9 sebelumnya, adalah sebagai berikut:

- Lapisan bawaan tidak lagi disediakan atau didukung untuk tumpukan Chef 12 untuk Linux. Karena hanya resep kustom Anda yang dijalankan, menghapus dukungan ini memberikan transparansi total tentang bagaimana instans diatur dan membuat buku masak khusus lebih mudah untuk ditulis dan dipelihara. Misalnya, tidak perlu lagi menimpa atribut resep AWS OpsWorks Stacks bawaan. Penghapusan lapisan bawaan juga memungkinkan AWS OpsWorks Stacks untuk mendukung buku masak yang dikembangkan dan dipelihara dengan lebih baik oleh komunitas Chef, sehingga Anda dapat memanfaatkannya sepenuhnya. Jenis lapisan bawaan yang tidak lagi tersedia di Chef 12 untuk Linux adalah: AWS Flow (Ruby), Ganglia, Java App Server HAProxy, Memcached, MySQL, Node.js App Server, PHP App Server, Rails AppServer, dan Static Web Server.
  - Karena AWS OpsWorks Stacks menjalankan resep yang Anda berikan, tidak perlu lagi mengganti atribut AWS OpsWorks Stacks bawaan dengan menjalankan buku masak khusus. Untuk mengganti atribut dalam resep Anda sendiri atau komunitas, ikuti petunjuk dan contoh di Tentang Atribut dalam dokumentasi Chef 12.
- AWS OpsWorks Stacks terus memberikan dukungan untuk lapisan berikut untuk tumpukan Chef 12 Linux:
  - Lapisan Kustom
  - Lapisan Layanan Amazon RDS
  - Lapisan Klaster ECS
- Konfigurasi tumpukan dan kantong data untuk Chef 12 Linux telah berubah agar terlihat sangat mirip dengan rekan-rekan mereka untuk Chef 12.2 Windows. Ini membuatnya lebih mudah untuk menanyakan, menganalisis, dan memecahkan masalah tas data ini, terutama jika Anda bekerja dengan tumpukan dengan jenis sistem operasi yang berbeda. Perhatikan bahwa AWS

OpsWorks Stacks tidak mendukung kantong data terenkripsi. Untuk menyimpan data sensitif dalam bentuk terenkripsi, seperti kata sandi atau sertifikat, sebaiknya simpan di bucket S3 pribadi. Anda kemudian dapat membuat resep khusus yang menggunakan Amazon SDK for Ruby untuk mengambil data. Sebagai contoh, lihat. Menggunakan SDK untuk Ruby Untuk informasi selengkapnya, lihatAWS OpsWorks Referensi Tas Data Stacks.

- Di Chef 12 Linux, Berkshelf tidak lagi diinstal pada instance stack. Sebagai gantinya, kami menyarankan Anda menggunakan Berkshelf pada mesin pengembangan lokal untuk mengemas dependensi buku masak Anda secara lokal. Kemudian unggah paket Anda, dengan dependensi yang disertakan, ke Amazon Simple Storage Service. Terakhir, ubah tumpukan Chef 12 Linux Anda untuk menggunakan paket yang diunggah sebagai sumber buku masak. Untuk informasi selengkapnya, lihat Ketergantungan Buku Masak Kemasan Secara Lokal.
- Konfigurasi RAID untuk volume EBS tidak lagi didukung. Untuk meningkatkan kinerja, Anda dapat menggunakan IOPS yang disediakan untuk Amazon Elastic Block Store (Amazon EBS).
- autofs tidak lagi didukung.
- Repositori subversi tidak lagi didukung.
- Instalasi paket OS per lapisan sekarang harus dilakukan dengan resep khusus. Untuk informasi selengkapnya, lihat Instalasi Package Per-layer.

Sistem Operasi yang Didukung

Chef 12 mendukung sistem operasi Linux yang sama dengan versi Chef sebelumnya. Untuk daftar jenis dan versi sistem operasi Linux yang dapat digunakan oleh tumpukan Chef 12 Linux, lihatSistem operasi Linux.

Tipe Instans Yang Didukung

AWS OpsWorks Stacks mendukung semua jenis instans untuk tumpukan Chef 12 Linux kecuali jenis instans khusus seperti komputasi klaster komputasi kinerja tinggi (HPC), GPU cluster, dan tipe instance cluster memori tinggi.

Informasi Selengkapnya

Untuk mempelajari lebih lanjut tentang cara bekerja dengan tumpukan Chef 12 untuk Linux, lihat berikut ini:

- Memulai: Contoh

Memperkenalkan Anda ke AWS OpsWorks Stacks dengan membimbing Anda melalui latihan langsung singkat dengan konsol AWS OpsWorks Stacks untuk membuat lingkungan aplikasi Node.js.

- Memulai: Linux

  Memperkenalkan Anda ke AWS OpsWorks Stacks dan Chef 12 Linux dengan membimbing Anda melalui latihan langsung dengan konsol AWS OpsWorks Stacks untuk membuat tumpukan Chef 12 Linux dasar yang berisi lapisan sederhana dengan aplikasi Node.js yang melayani lalu lintas.

- Lapisan Kustom

  Memberikan panduan untuk menambahkan lapisan yang berisi buku masak dan resep ke tumpukan Chef 12 Linux. Anda dapat menggunakan buku masak dan resep yang tersedia yang disediakan komunitas Chef, atau Anda dapat membuatnya sendiri.

- Pindah ke Data Bags

  Membandingkan dan membedakan instance JSON yang digunakan oleh tumpukan Linux yang menjalankan Chef 11 dan versi sebelumnya dengan Chef 12. Juga menyediakan petunjuk ke dokumentasi referensi untuk format JSON instance Chef 12.

Memindahkan Pengaturan Stack dari Atribut ke Data Bags

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks memaparkan berbagai macam pengaturan tumpukan ke resep Chef Anda. Pengaturan tumpukan ini mencakup nilai-nilai seperti:

- Sumber buku masak tumpukan URLs
- Konfigurasi volume lapisan
- Nama host instance

- Nama DNS Elastic Load Balancing
- Sumber aplikasi URLs
- Nama pengguna

Merujuk pengaturan tumpukan dari resep membuat kode resep lebih kuat dan lebih sedikit rawan kesalahan daripada pengaturan tumpukan hard-coding langsung dalam resep. Topik ini menjelaskan cara mengakses pengaturan tumpukan ini serta cara berpindah dari atribut di Chef 11.10 dan versi sebelumnya untuk Linux ke kantong data di Chef 12 Linux.

Di Chef 11.10 dan versi sebelumnya untuk Linux, pengaturan tumpukan tersedia sebagai atribut Chef dan diakses melalui `node` objek Chef atau melalui pencarian Chef. Atribut ini disimpan pada instance AWS OpsWorks Stacks dalam satu set file JSON di direktori. `/var/lib/aws/opsworks/chef` Untuk informasi selengkapnya, lihat Konfigurasi Stack dan Atribut Deployment: Linux.

Di Chef 12 Linux, pengaturan tumpukan tersedia sebagai kantong data Chef dan hanya diakses melalui pencarian Chef. Kantong data disimpan pada instance AWS OpsWorks Stacks dalam satu set file JSON di `/var/chef/runs/`*`run-ID`*`/data_bags` direktori, di mana *`run-ID`* adalah ID unik yang ditetapkan AWS OpsWorks Stacks ke setiap Chef yang dijalankan pada sebuah instance. Pengaturan tumpukan tidak lagi tersedia sebagai atribut Chef, sehingga pengaturan tumpukan tidak dapat lagi diakses melalui `node` objek Chef. Untuk informasi selengkapnya, lihat AWS OpsWorks Referensi Tas Data Stacks.

Misalnya, di Chef 11.10 dan versi sebelumnya untuk Linux, kode resep berikut menggunakan `node` objek Chef untuk mendapatkan atribut yang mewakili nama pendek aplikasi dan URL sumber. Kemudian menggunakan log Chef untuk menulis dua nilai atribut ini:

```
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']}' **********")
```

Di Chef 12 Linux, kode resep berikut menggunakan indeks `aws_opsworks_app` pencarian untuk mendapatkan konten item tas data pertama di tas `aws_opsworks_app` data. Kode kemudian menulis dua pesan ke log Chef, satu dengan konten tas data nama pendek aplikasi, dan satu lagi dengan konten tas data URL sumber aplikasi:

```
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']}' **********")
```

Untuk memigrasikan kode resep Anda yang mengakses pengaturan tumpukan dari Chef 11.10 dan versi sebelumnya untuk Linux ke Chef 12 Linux, Anda harus merevisi kode Anda ke:

- Akses tas data Chef alih-alih atribut Chef.
- Gunakan pencarian Chef alih-alih `node` objek Chef.
- Gunakan nama tas data AWS OpsWorks Stacks seperti`aws_opsworks_app`, alih-alih menggunakan nama atribut AWS OpsWorks Stacks seperti `opsworks` dan. `deploy`

Untuk informasi selengkapnya, lihat AWS OpsWorks Referensi Tas Data Stacks.

Support untuk Versi Chef Sebelumnya di AWS OpsWorks Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini memberikan gambaran singkat tentang dokumentasi AWS OpsWorks Stacks untuk versi Chef sebelumnya.

Chef 11.10 dan Versi Sebelumnya untuk Linux

Menyediakan dokumentasi tentang dukungan AWS OpsWorks Stacks untuk Chef 11.10, 11.4, dan 0.9 untuk tumpukan Linux.

Chef 11.10 dan Versi Sebelumnya untuk Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini memberikan gambaran singkat tentang dokumentasi AWS OpsWorks Stacks untuk Chef 11.10, 11.4, dan 0.9 untuk Linux.

Memulai dengan Chef 11 Linux Stacks

Menyediakan panduan yang menunjukkan kepada Anda cara membuat tumpukan server aplikasi PHP yang sederhana namun fungsional.

Membuat Stack Node.js Pertama Anda

Menjelaskan cara membuat tumpukan Linux yang mendukung server aplikasi Node.js dan cara menerapkan aplikasi sederhana.

Menyesuaikan Tumpukan AWS OpsWorks

Menjelaskan cara menyesuaikan AWS OpsWorks Tumpukan untuk memenuhi kebutuhan spesifik Anda.

Buku masak 101

Menjelaskan cara menerapkan resep untuk instance AWS OpsWorks Stacks.

Load Balancing sebuah Layer

Menjelaskan cara menggunakan opsi penyeimbangan beban AWS OpsWorks Stacks yang tersedia.

Menjalankan Stack di VPC

Menjelaskan cara membuat dan menjalankan tumpukan di cloud pribadi virtual.

Migrasi dari Chef Server

Memberikan panduan untuk bermigrasi dari Chef Server ke AWS OpsWorks Stacks.

[AWS OpsWorks Referensi Lapisan Tumpukan]

Menjelaskan lapisan bawaan AWS OpsWorks Stacks yang tersedia.

[Komponen Buku Masak]

Menjelaskan tiga komponen buku masak standar: atribut, templat, dan resep.

[Konfigurasi Stack dan Atribut Deployment: Linux]

Menjelaskan konfigurasi tumpukan dan atribut penyebaran untuk Linux.

[Atribut Buku Masak Bawaan]

Menjelaskan cara menggunakan atribut resep bawaan untuk mengontrol konfigurasi perangkat lunak yang diinstal.

[Pemecahan Masalah Chef 11.10 dan Versi Sebelumnya untuk Linux]

Menjelaskan pendekatan untuk memecahkan masalah berbagai masalah di AWS OpsWorks Stacks.

Memulai dengan Chef 11 Linux Stacks

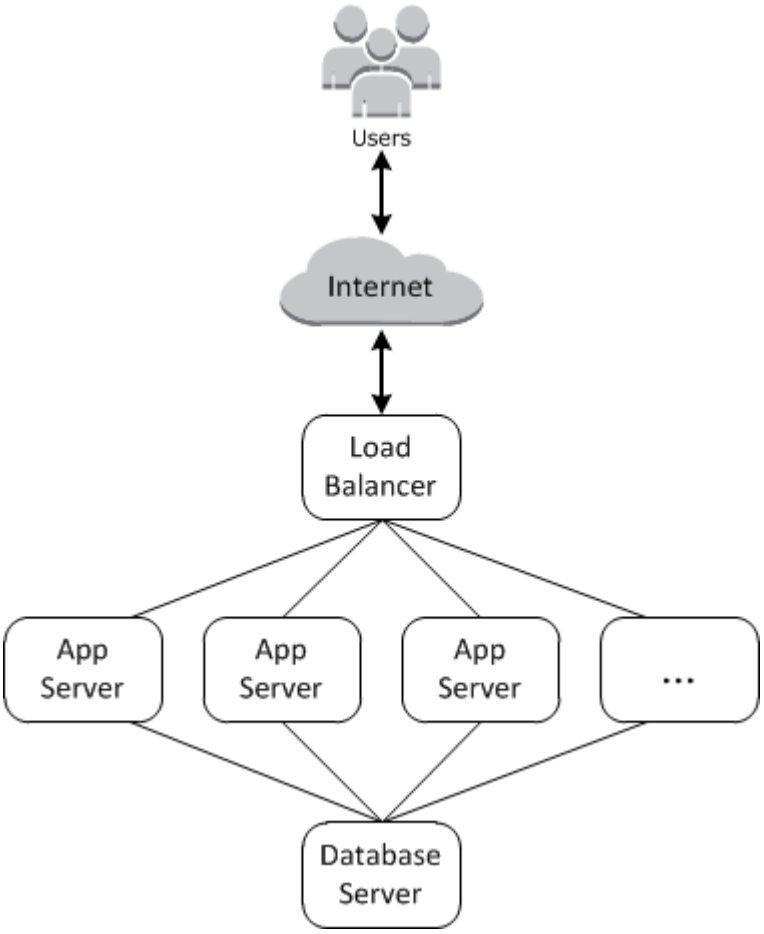
Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Bagian ini menjelaskan cara memulai dengan tumpukan Linux menggunakan Chef 11. Untuk informasi tentang memulai dengan tumpukan Chef 12 Linux, lihatMemulai: Linux. Untuk informasi tentang memulai dengan tumpukan Chef 12 Windows, lihatMemulai: Windows.

Aplikasi berbasis cloud biasanya memerlukan sekelompok sumber daya terkait — server aplikasi, server basis data, dan sebagainya — yang harus dibuat dan dikelola secara kolektif. Kumpulan instance ini disebut tumpukan. Tumpukan aplikasi sederhana mungkin terlihat seperti berikut ini.



Arsitektur dasar terdiri dari yang berikut:

- Load balancer untuk mendistribusikan lalu lintas masuk dari pengguna secara merata di seluruh server aplikasi.
- Satu set instance server aplikasi, sebanyak yang diperlukan untuk menangani lalu lintas.
- Server database untuk menyediakan server aplikasi dengan penyimpanan data back-end.

Selain itu, Anda biasanya memerlukan cara untuk mendistribusikan aplikasi ke server aplikasi, memantau tumpukan, dan sebagainya.

AWS OpsWorks Stacks menyediakan cara sederhana dan mudah untuk membuat dan mengelola tumpukan dan aplikasi serta sumber daya terkait. Bab ini memperkenalkan dasar-dasar AWS OpsWorks tumpukan — bersama dengan beberapa fitur yang lebih canggih — dengan memandu

Anda melalui proses pembuatan tumpukan server aplikasi dalam diagram. Ini menggunakan model pengembangan inkremental yang memudahkan AWS OpsWorks Stacks untuk diikuti: Siapkan tumpukan dasar dan, setelah berfungsi dengan benar, tambahkan komponen hingga Anda tiba pada implementasi berfitur lengkap.

- Langkah 1: Lengkapi Prasyaratmenunjukkan cara mengatur untuk memulai penelusuran.
- Langkah 2: Buat Tumpukan Server Aplikasi Sederhana - Chef 11menunjukkan cara membuat tumpukan minimal yang terdiri dari satu server aplikasi.
- Langkah 3: Tambahkan Toko Data Back-endmenunjukkan cara menambahkan server database dan menghubungkannya ke server aplikasi.
- Langkah 4: Scale Out MyStackmenunjukkan cara menskalakan tumpukan untuk menangani peningkatan beban dengan menambahkan lebih banyak server aplikasi, dan penyeimbang beban untuk mendistribusikan lalu lintas yang masuk.

Topik

- Langkah 1: Lengkapi Prasyarat
- Langkah 2: Buat Tumpukan Server Aplikasi Sederhana - Chef 11
- Langkah 3: Tambahkan Toko Data Back-end
- Langkah 4: Scale Out MyStack
- Langkah 5: Hapus MyStack

Langkah 1: Lengkapi Prasyarat

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Selesaikan langkah-langkah pengaturan berikut sebelum Anda dapat memulai penelusuran. Langkah-langkah penyiapan ini termasuk mendaftar untuk AWS akun, membuat pengguna administratif, dan menetapkan izin akses ke Stacks. AWS OpsWorks

Jika Anda telah menyelesaikan salah satu Memulai dengan AWS OpsWorks Stacks penelusuran, maka Anda telah memenuhi prasyarat untuk panduan ini, dan Anda dapat melompat ke depan. Langkah 2: Buat Tumpukan Server Aplikasi Sederhana - Chef 11

Topik

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif
- Tetapkan Izin Akses Layanan ke Pengguna Anda

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Tetapkan Izin Akses Layanan ke Pengguna Anda

Aktifkan akses ke layanan AWS OpsWorks Stacks (dan layanan terkait yang diandalkan oleh AWS OpsWorks Stacks) dengan menambahkan `AWSOpsWorks_FullAccess` dan `AmazonS3FullAccess` izin ke peran atau pengguna Anda.

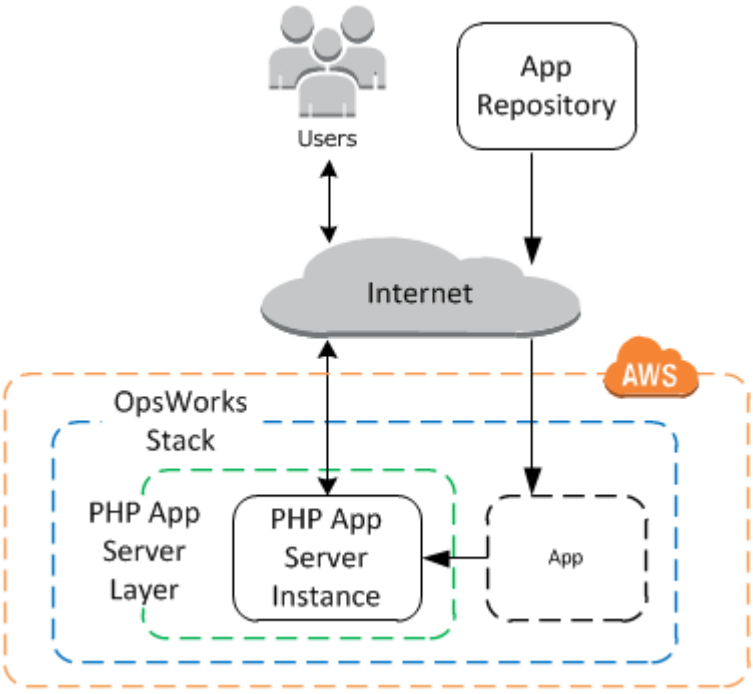
Untuk informasi selengkapnya tentang menambahkan izin, lihat Menambahkan izin identitas IAM (konsol).

Anda sekarang telah menyelesaikan semua langkah pengaturan dan dapat memulai penelusuran ini.

Langkah 2: Buat Tumpukan Server Aplikasi Sederhana - Chef 11

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan server aplikasi dasar terdiri dari satu instance server aplikasi dengan alamat IP publik untuk menerima permintaan pengguna. Kode aplikasi dan file terkait disimpan dalam repositori terpisah dan digunakan dari sana ke server. Diagram berikut menggambarkan tumpukan seperti itu.



Tumpukan memiliki komponen-komponen berikut:

- Lapisan, yang mewakili sekelompok instance dan menentukan bagaimana mereka akan dikonfigurasi.

  Lapisan dalam contoh ini mewakili sekelompok instance PHP App Server.
- Sebuah instance, yang mewakili EC2 instance Amazon.

  Dalam hal ini, instance dikonfigurasi untuk menjalankan server aplikasi PHP. Lapisan dapat memiliki sejumlah contoh. AWS OpsWorks Stacks juga mendukung beberapa server aplikasi lainnya. Untuk informasi selengkapnya, lihat Lapisan Server Aplikasi.
- Aplikasi, yang berisi informasi yang diperlukan untuk menginstal aplikasi di server aplikasi.

  Kode disimpan dalam repositori jarak jauh, seperti repositori Git atau bucket Amazon S3.

Bagian berikut menjelaskan cara menggunakan konsol AWS OpsWorks Stacks untuk membuat tumpukan dan menyebarkan aplikasi. Anda juga dapat menggunakan AWS CloudFormation template untuk menyediakan tumpukan. Untuk contoh templat yang menyediakan tumpukan yang dijelaskan dalam topik ini, lihat OpsWorks Cuplikan AWS.

Topik

- Langkah 2.1: Buat Stack - Chef 11

- Langkah 2.2: Tambahkan Lapisan Server Aplikasi PHP - Chef 11
- Langkah 2.3: Tambahkan Instance ke PHP App Server Layer - Chef 11
- Langkah 2.4: Membuat dan Menyebarkan Aplikasi - Chef 11

Langkah 2.1: Buat Stack - Chef 11

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda memulai proyek AWS OpsWorks Stacks dengan membuat tumpukan, yang bertindak sebagai wadah untuk instance Anda dan sumber daya lainnya. Konfigurasi tumpukan menentukan beberapa pengaturan dasar, seperti wilayah AWS dan sistem operasi default, yang dibagikan oleh semua instance tumpukan.

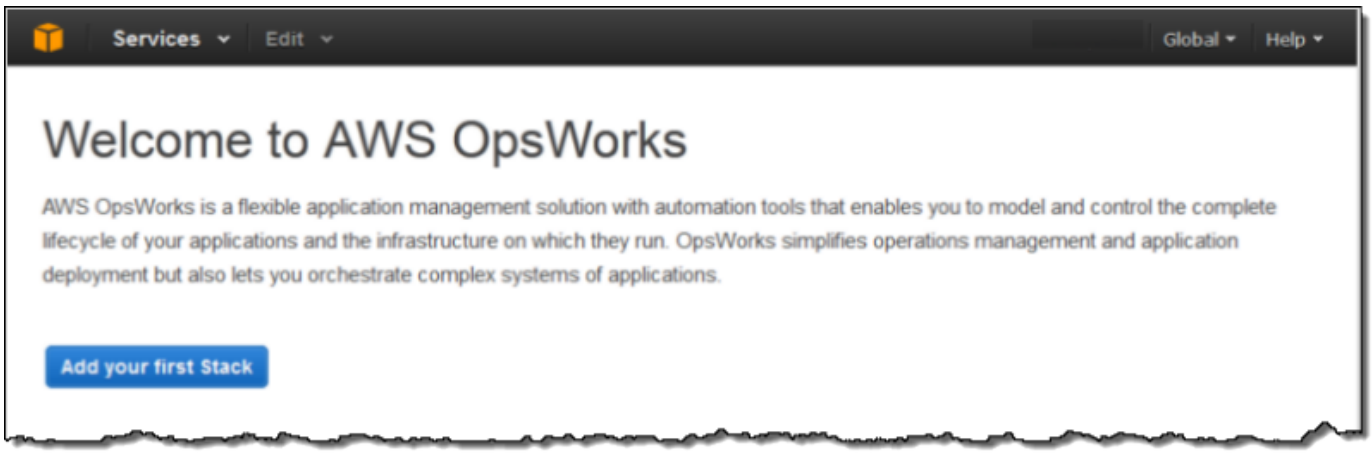
> **Note**
>
> Halaman ini membantu Anda membuat tumpukan Chef 11. Untuk informasi tentang cara membuat tumpukan Chef 12, lihat Membuat Tumpukan.

Halaman ini membantu Anda membuat tumpukan di Chef 11.

Untuk membuat tumpukan baru

1. Tambahkan Stack

   Masuk ke konsol AWS OpsWorks Stacks. Jika akun tidak memiliki tumpukan yang ada, Anda akan melihat OpsWorks halaman Selamat Datang di AWS; klik Tambahkan tumpukan pertama Anda. Jika tidak, Anda akan melihat dasbor AWS OpsWorks Stacks, yang mencantumkan tumpukan akun Anda; klik Tambahkan Tumpukan.



2. Konfigurasikan Stack

   Pada halaman Add Stack, pilih tumpukan Chef 11 dan tentukan pengaturan berikut:

   Nama tumpukan

   Masukkan nama untuk tumpukan Anda, yang dapat berisi karakter alfanumerik (a—z, A—Z, dan 0-9), dan tanda hubung (-). Contoh tumpukan untuk panduan ini diberi nama. **`MyStack`**

   Wilayah

   Pilih US West (Oregon) sebagai wilayah tumpukan.

   Terima nilai default untuk pengaturan lain dan klik Tambah Tumpukan. Untuk informasi lebih lanjut tentang berbagai pengaturan tumpukan, lihatMembuat Tumpukan Baru.

Langkah 2.2: Tambahkan Lapisan Server Aplikasi PHP - Chef 11

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Meskipun tumpukan pada dasarnya adalah wadah untuk instance, Anda tidak menambahkan instance langsung ke tumpukan. Anda menambahkan layer, yang mewakili sekelompok instance terkait, dan kemudian menambahkan instance ke layer.

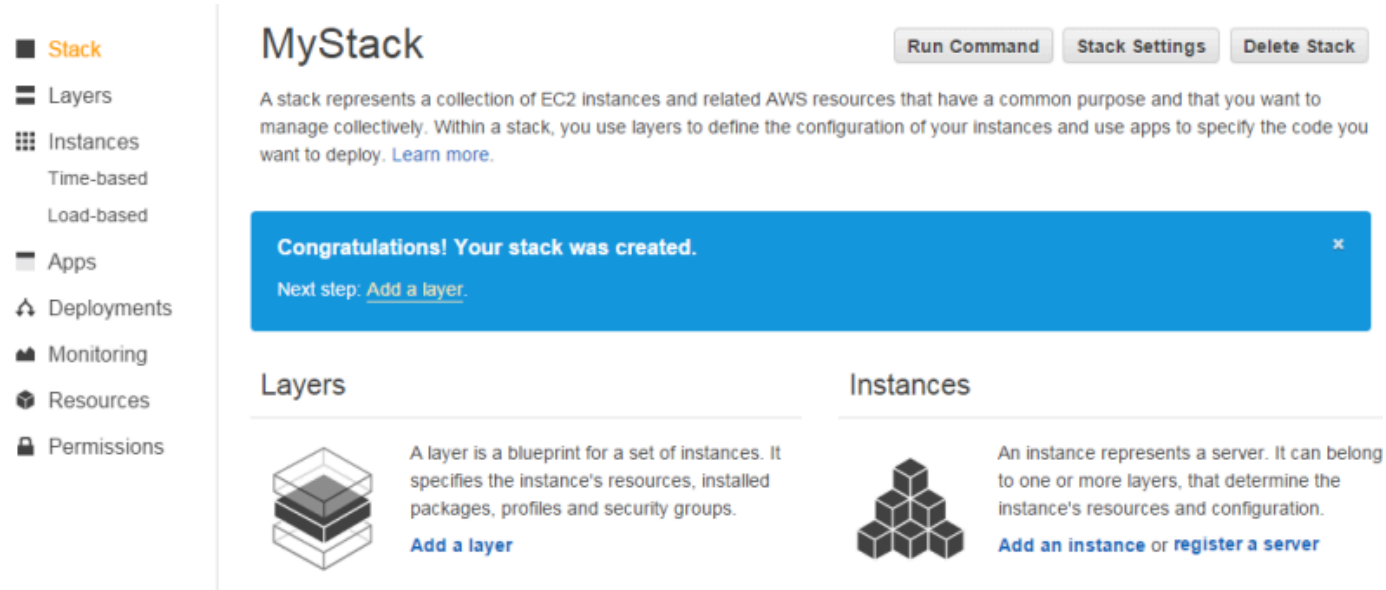
Lapisan pada dasarnya adalah cetak biru yang digunakan AWS OpsWorks Stacks untuk membuat kumpulan EC2 instance Amazon dengan konfigurasi yang sama. Anda menambahkan satu lapisan ke tumpukan untuk setiap grup instance terkait. AWS OpsWorks Stacks mencakup satu set lapisan built-in untuk mewakili kelompok instance yang menjalankan paket perangkat lunak standar seperti server database MySQL atau server aplikasi PHP. Selain itu, Anda dapat membuat lapisan yang disesuaikan sebagian atau sepenuhnya sesuai dengan kebutuhan spesifik Anda. Untuk informasi selengkapnya, lihat Menyesuaikan Tumpukan AWS OpsWorks.

MyStack memiliki satu lapisan, lapisan PHP App Server bawaan, yang mewakili sekelompok contoh yang berfungsi sebagai server aplikasi PHP. Untuk informasi selengkapnya, termasuk deskripsi lapisan bawaan, lihatLapisan.

Untuk menambahkan lapisan PHP App Server ke MyStack

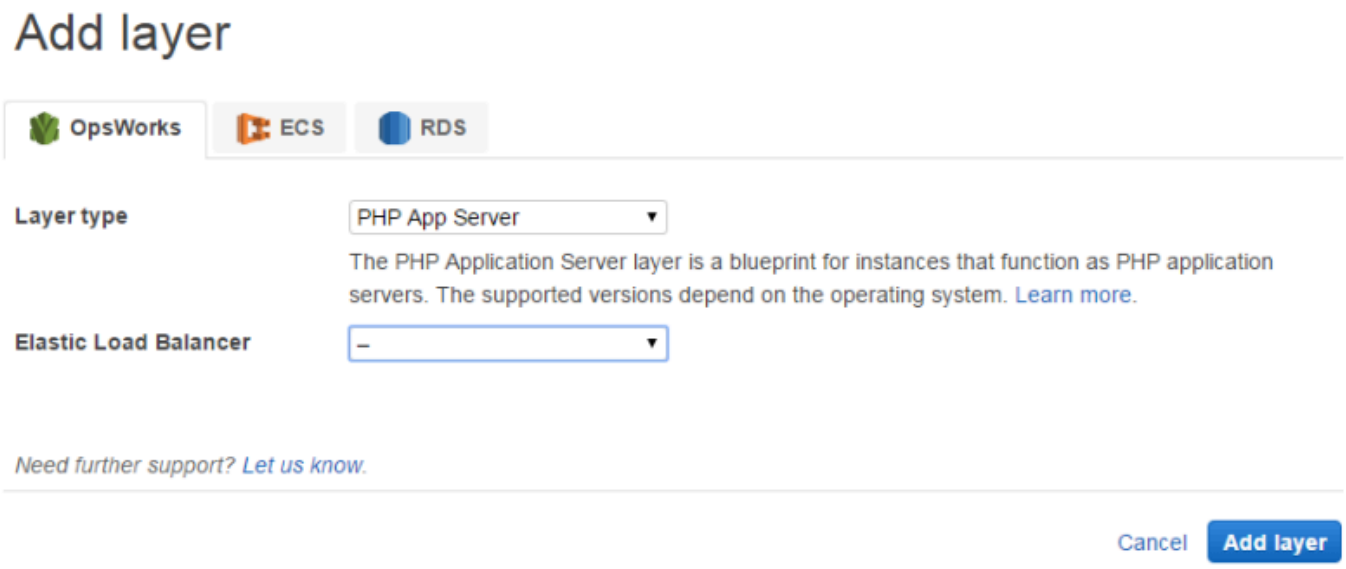
1. Buka Halaman Add Layer

   Setelah Anda selesai membuat tumpukan, AWS OpsWorks Stacks menampilkan halaman Stack. Klik Add a layer untuk menambahkan layer pertama Anda.



2. Tentukan Tipe Layer dan Konfigurasikan Layer

   Di kotak tipe Layer, pilih PHP App Server, terima pengaturan Elastic Load Balancer default dan klik Add Layer. Setelah Anda membuat layer, Anda dapat menentukan atribut lain seperti konfigurasi volume EBS dengan mengedit layer.



Langkah 2.3: Tambahkan Instance ke PHP App Server Layer - Chef 11

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Instans AWS OpsWorks Stacks mewakili EC2 instance Amazon tertentu:

- Konfigurasi instans menentukan beberapa dasar-dasar seperti EC2operating sistem dan ukuran Amazon; itu berjalan tetapi tidak melakukan banyak hal.
- Lapisan instance menambahkan fungsionalitas ke instance dengan menentukan paket mana yang akan diinstal, apakah instance memiliki alamat IP Elastis, dan sebagainya.

AWS OpsWorks Stacks menginstal agen pada setiap instance yang berinteraksi dengan layanan. Untuk menambahkan fungsionalitas lapisan ke sebuah instance, AWS OpsWorks Stacks mengarahkan agen untuk menjalankan aplikasi kecil yang disebut resep Chef, yang dapat menginstal aplikasi dan paket, membuat file konfigurasi, dan sebagainya. AWS OpsWorks Stacks menjalankan resep pada titik-titik penting dalam siklus hidup instance. Misalnya, OpsWorks menjalankan resep Setup setelah instance selesai booting untuk menangani tugas seperti menginstal perangkat lunak, dan menjalankan Deploy recipes saat Anda menerapkan aplikasi untuk menginstal kode dan file terkait.

> **Note**
>
> Jika Anda ingin tahu tentang cara kerja resep, semua resep bawaan AWS OpsWorks Stacks ada di GitHub repositori publik: Buku Masak. OpsWorks Anda juga dapat membuat resep kustom Anda sendiri dan memiliki AWS OpsWorks Stacks menjalankannya, seperti yang dijelaskan nanti.

Untuk menambahkan server aplikasi PHP MyStack, tambahkan instance ke lapisan PHP App Server yang Anda buat pada langkah sebelumnya.

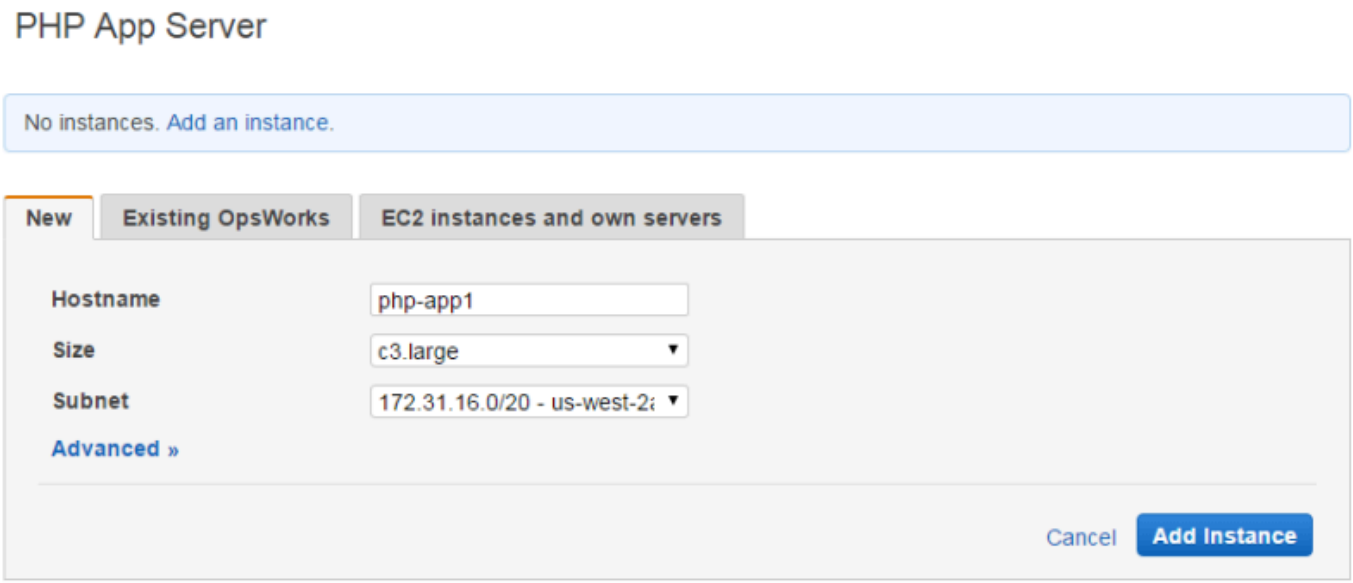
Untuk menambahkan instance ke lapisan PHP App Server

1. Buka Tambahkan Instance

   Setelah Anda selesai menambahkan layer, AWS OpsWorks Stacks menampilkan halaman Layers. Klik Instans di panel navigasi dan di bawah PHP App Server, klik Tambahkan instance.

2. Konfigurasikan Instance

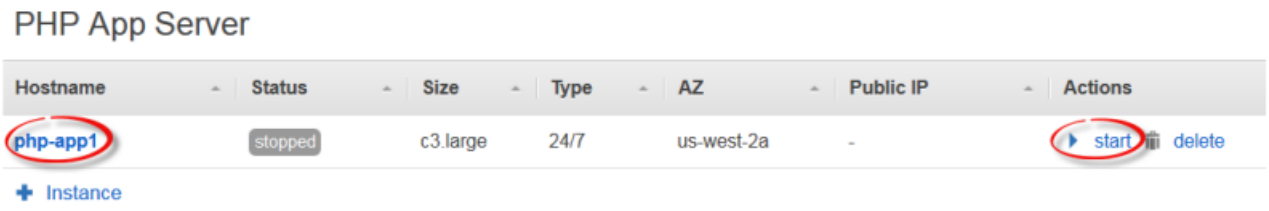
   Setiap instance memiliki nama host default yang dibuat untuk Anda oleh AWS OpsWorks Stacks. Dalam contoh ini, AWS OpsWorks Stacks hanya menambahkan angka ke nama pendek layer. Anda dapat mengonfigurasi setiap instance secara terpisah, termasuk mengganti beberapa pengaturan default yang Anda tentukan saat membuat tumpukan, seperti Availability Zone atau sistem operasi. Untuk panduan ini, cukup terima pengaturan default dan klik Add Instance untuk menambahkan instance ke layer. Untuk informasi selengkapnya, lihat Instans.



3. Mulai Instance

   Sejauh ini, Anda baru saja menentukan konfigurasi instance. Anda harus memulai instance untuk membuat EC2 instance Amazon yang sedang berjalan. AWS OpsWorks Stacks kemudian menggunakan pengaturan konfigurasi untuk meluncurkan EC2 instance Amazon di Availability Zone yang ditentukan. Detail bagaimana Anda memulai instance bergantung pada jenis penskalaan instans. Pada langkah sebelumnya, Anda membuat instance dengan tipe penskalaan default, 24/7, yang harus dimulai secara manual dan kemudian dijalankan hingga dihentikan secara manual. Anda juga dapat membuat jenis penskalaan berbasis waktu dan berbasis beban, yang AWS OpsWorks Stacks secara otomatis dimulai dan dihentikan berdasarkan jadwal atau beban saat ini. Untuk informasi selengkapnya, lihat Mengelola beban dengan instans berbasis waktu dan berbasis beban.

   Pergi ke php-app1 di bawah PHP App Server dan klik mulai di kolom Actions baris untuk memulai instance.

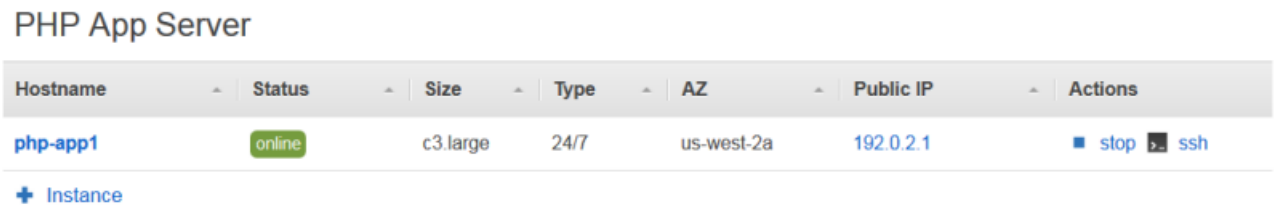


4. Pantau Status Instans selama Startup

   Biasanya diperlukan beberapa menit untuk mem-boot EC2 instance Amazon dan menginstal paket. Saat startup berlangsung, bidang Status instance menampilkan rangkaian nilai berikut:

1. request - AWS OpsWorks Stacks telah memanggil EC2 layanan Amazon untuk membuat EC2 instance Amazon.
2. tertunda - AWS OpsWorks Tumpukan sedang menunggu EC2 instans Amazon dimulai.
3. booting - EC2 Instans Amazon sedang booting.
4. running_setup - Agen AWS OpsWorks Stacks menjalankan resep Pengaturan lapisan, yang menangani tugas-tugas seperti mengonfigurasi dan menginstal paket, dan resep Deploy, yang menyebarkan aplikasi apa pun ke instance.
5. online - Instans siap digunakan.

Setelah php-app1 online, halaman Instances akan terlihat seperti berikut:



Halaman dimulai dengan ringkasan cepat dari semua instance tumpukan Anda. Saat ini, ini menunjukkan satu contoh online. Di kolom php-app1 Actions, perhatikan bahwa stop, yang menghentikan instance, telah menggantikan start dan delete.

Langkah 2.4: Membuat dan Menyebarkan Aplikasi - Chef 11

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Agar MyStack lebih berguna, Anda perlu menerapkan aplikasi ke instance PHP App Server. Anda menyimpan kode aplikasi dan file terkait dalam repositori, seperti Git. Anda perlu mengambil beberapa langkah untuk mendapatkan file-file tersebut ke server aplikasi Anda:

> **Note**
>
> Prosedur di bagian ini berlaku untuk tumpukan Chef 11. Untuk informasi tentang cara menambahkan aplikasi ke layer di tumpukan Chef 12, lihatMenambahkan Aplikasi.

1. Buat aplikasi.

   Aplikasi berisi informasi yang dibutuhkan AWS OpsWorks Stacks untuk mengunduh kode dan file terkait dari repositori. Anda juga dapat menentukan informasi tambahan seperti domain aplikasi.
2. Menerapkan aplikasi ke server aplikasi Anda.

   Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks akan memicu peristiwa siklus hidup Deploy. Agen kemudian menjalankan resep Deploy instance, yang mengunduh file ke direktori yang sesuai bersama dengan tugas terkait seperti mengonfigurasi server, memulai ulang layanan, dan sebagainya.

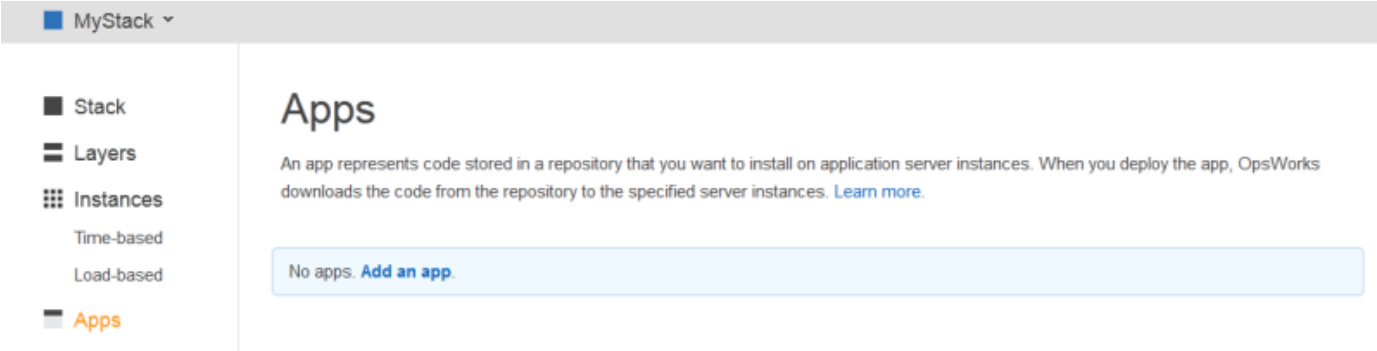
> **Note**
>
> Saat Anda membuat instance baru, AWS OpsWorks Stacks secara otomatis akan menerapkan aplikasi apa pun yang ada ke instans. Namun, saat membuat aplikasi baru atau memperbarui aplikasi yang sudah ada, Anda harus menerapkan aplikasi secara manual atau memperbarui ke semua instance yang ada.

Langkah ini menunjukkan cara menerapkan aplikasi contoh secara manual dari repositori Git publik ke server aplikasi. Jika Anda ingin memeriksa aplikasi, buka https://github.com/amazonwebservices/opsworks-demo-php-simple-app. Aplikasi yang digunakan dalam contoh ini ada di cabang version1. AWS OpsWorks Stacks juga mendukung beberapa jenis repositori lainnya. Untuk informasi selengkapnya, lihat Sumber Aplikasi.

Untuk membuat dan menerapkan aplikasi

1. Buka Halaman Aplikasi

   Di panel navigasi, klik Aplikasi dan pada halaman Aplikasi, klik Tambahkan aplikasi.



2. Konfigurasikan Aplikasi

   Pada halaman Aplikasi, tentukan nilai berikut:

   Nama

   Nama aplikasi, yang digunakan AWS OpsWorks Stacks untuk tujuan tampilan. Aplikasi contoh diberi nama**`SimplePHPApp`**. AWS OpsWorks Stacks juga menghasilkan nama pendek—simplephpapp untuk contoh ini—yang digunakan secara internal dan oleh resep Deploy, seperti yang dijelaskan nanti.

   Tipe

   Jenis aplikasi, yang menentukan tempat untuk menerapkan aplikasi. Contoh menggunakan PHP, yang menyebarkan aplikasi ke instance PHP App Server.

   Jenis sumber data

   Server basis data terkait. Untuk saat ini, pilih None; kami akan memperkenalkan server database diLangkah 3: Tambahkan Toko Data Back-end.

   Tipe repositori

   Jenis repositori aplikasi. Contoh aplikasi disimpan dalam repositori Git.

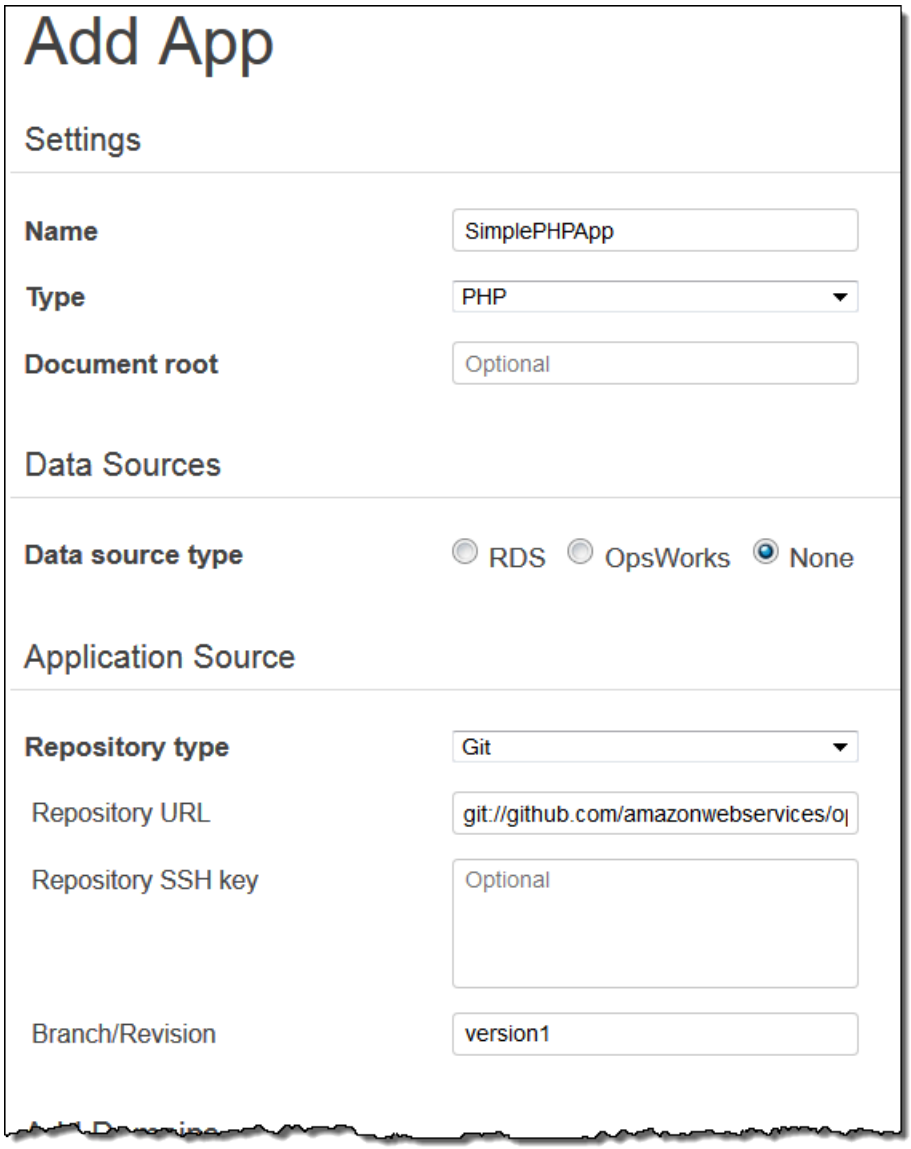
   URL Repositori

   URL repositori aplikasi. Contoh URL adalah: **`git://github.com/awslabs/opsworks-demo-php-simple-app.git`**

   Cabang/Revisi

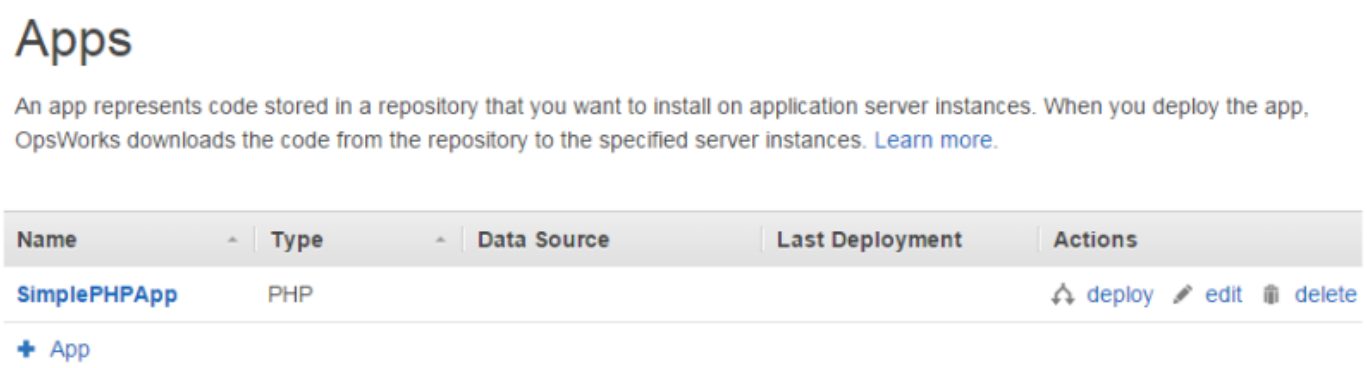
   Cabang atau versi aplikasi. Bagian dari panduan ini menggunakan cabang. **`version1`**

Simpan nilai default untuk setelan yang tersisa dan klik Tambah Aplikasi. Untuk informasi selengkapnya, lihat Menambahkan Aplikasi.



3. Buka Halaman Deployment

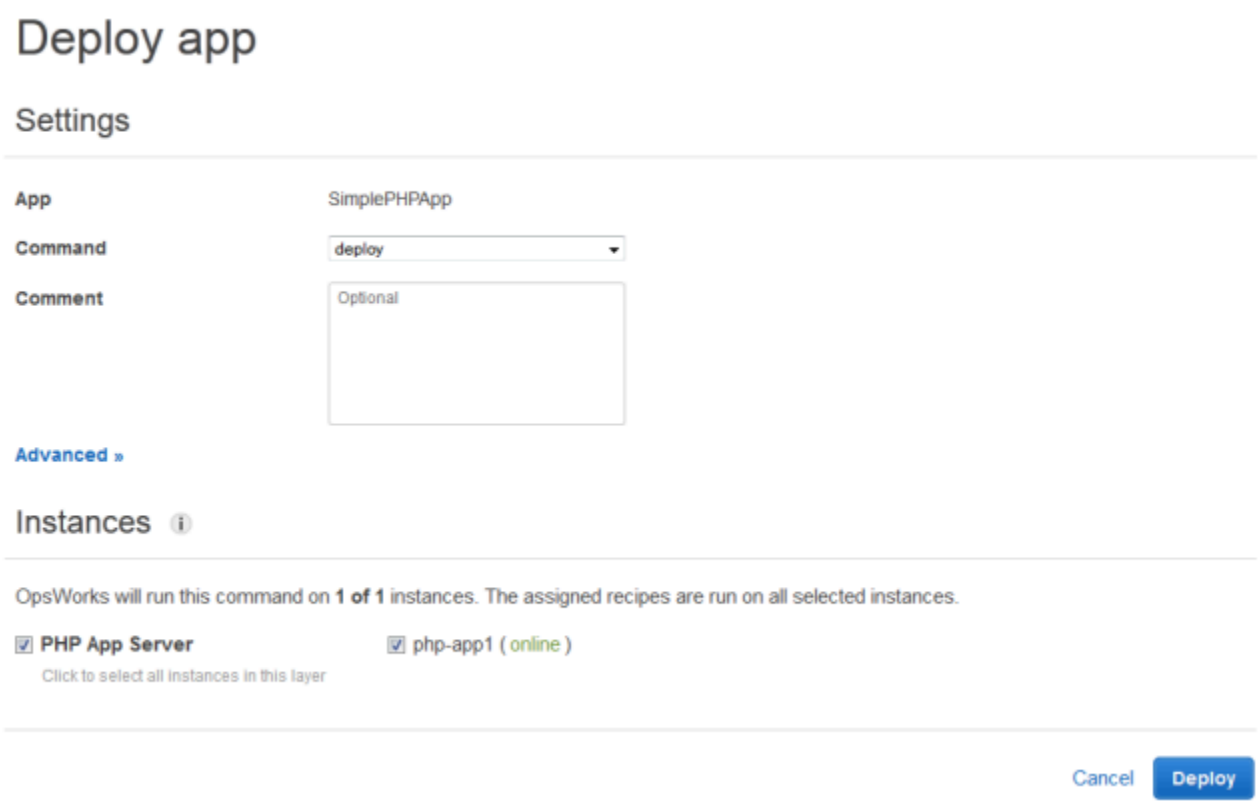
   Untuk menginstal kode di server, Anda harus menerapkan aplikasi. Untuk melakukannya, klik deploy di kolom PHPApp Tindakan Sederhana.



4. Menerapkan Aplikasi

   Saat Anda menerapkan aplikasi, agen menjalankan resep Deploy pada instance PHP App Server, yang mengunduh dan mengonfigurasi aplikasi.

   Perintah harus sudah diatur untuk menyebarkan. Simpan default untuk pengaturan lain dan klik Deploy untuk menyebarkan aplikasi.



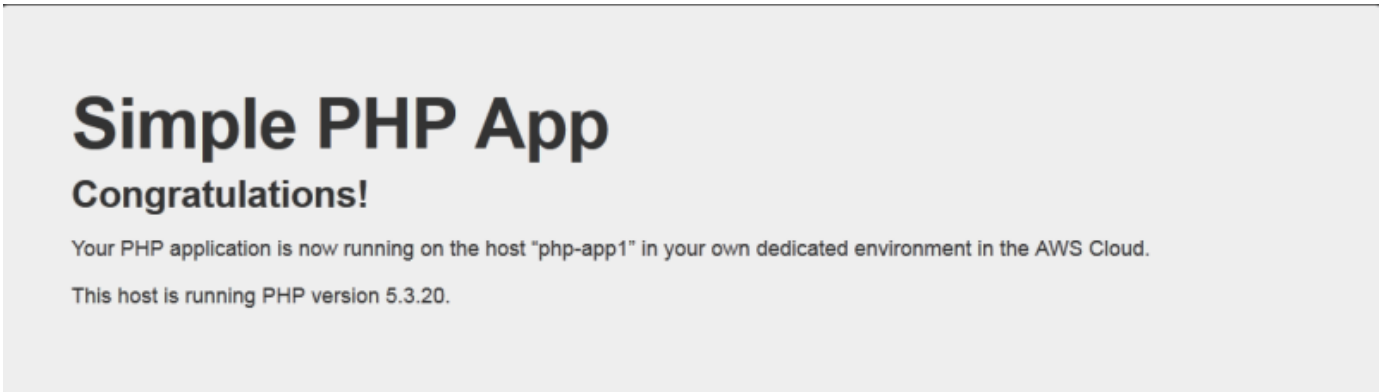
   Saat penerapan selesai, halaman Deployment menampilkan Status Sukses, dan php-app1 akan memiliki tanda centang hijau di sebelahnya.

5. Jalankan Sederhana PHPApp

PHPApp Sederhana sekarang diinstal dan siap digunakan. Untuk menjalankannya, klik Instans di panel navigasi untuk membuka halaman Instans. Kemudian klik alamat IP publik instans php-app1.



Anda akan melihat halaman seperti berikut di browser Anda.

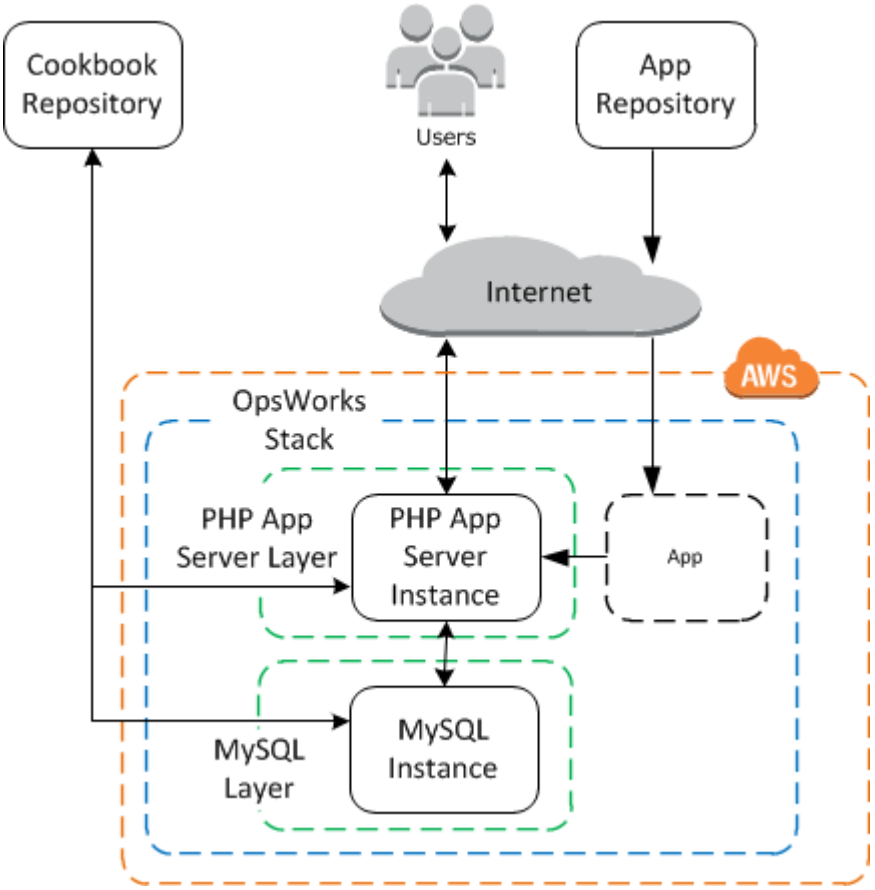


> **Note**
>
> Panduan ini mengasumsikan bahwa Anda akan melanjutkan ke bagian berikutnya dan akhirnya menyelesaikan seluruh panduan dalam satu sesi. Jika mau, Anda dapat berhenti kapan saja dan melanjutkan nanti dengan masuk ke AWS OpsWorks Stacks dan membuka tumpukan. Namun, Anda dikenakan biaya untuk semua sumber daya AWS yang Anda gunakan, seperti instans online. Untuk menghindari biaya yang tidak perlu, Anda dapat menghentikan instans Anda, yang mengakhiri EC2 instance yang sesuai. Anda dapat memulai instance lagi ketika Anda siap untuk melanjutkan.

Langkah 3: Tambahkan Toko Data Back-end

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Langkah 2.1: Buat Stack - Chef 11menunjukkan cara membuat tumpukan yang melayani aplikasi PHP. Namun, itu adalah aplikasi yang sangat sederhana yang tidak lebih dari menampilkan beberapa teks statis. Aplikasi produksi biasanya menggunakan penyimpanan data back-end, menghasilkan konfigurasi tumpukan sesuatu seperti ilustrasi berikut.



Bagian ini menunjukkan cara memperluas MyStack untuk menyertakan server database MySQL back-end. Anda perlu melakukan lebih dari sekadar menambahkan server MySQL ke tumpukan. Anda juga harus mengonfigurasi aplikasi untuk berkomunikasi dengan benar dengan server

database. AWS OpsWorks Tumpukan tidak melakukan ini untuk Anda; Anda perlu menerapkan beberapa resep khusus untuk menangani tugas itu.

Topik

- Langkah 3.1: Tambahkan Database Back-end
- Langkah 3.2: Perbarui Sederhana PHPApp
- Penyimpangan Singkat: Buku Masak, Resep, dan Atribut Tumpukan AWS OpsWorks
- Langkah 3.3: Tambahkan Buku Masak Kustom ke MyStack
- Langkah 3.4: Jalankan Resep
- Langkah 3.5: Menyebarkan SederhanaPHPApp, Versi 2
- Langkah 3.6: Jalankan Sederhana PHPApp

Langkah 3.1: Tambahkan Database Back-end

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Versi baru Simple PHPApp menyimpan datanya dalam database back-end. AWS OpsWorks Stacks mendukung dua jenis server database:

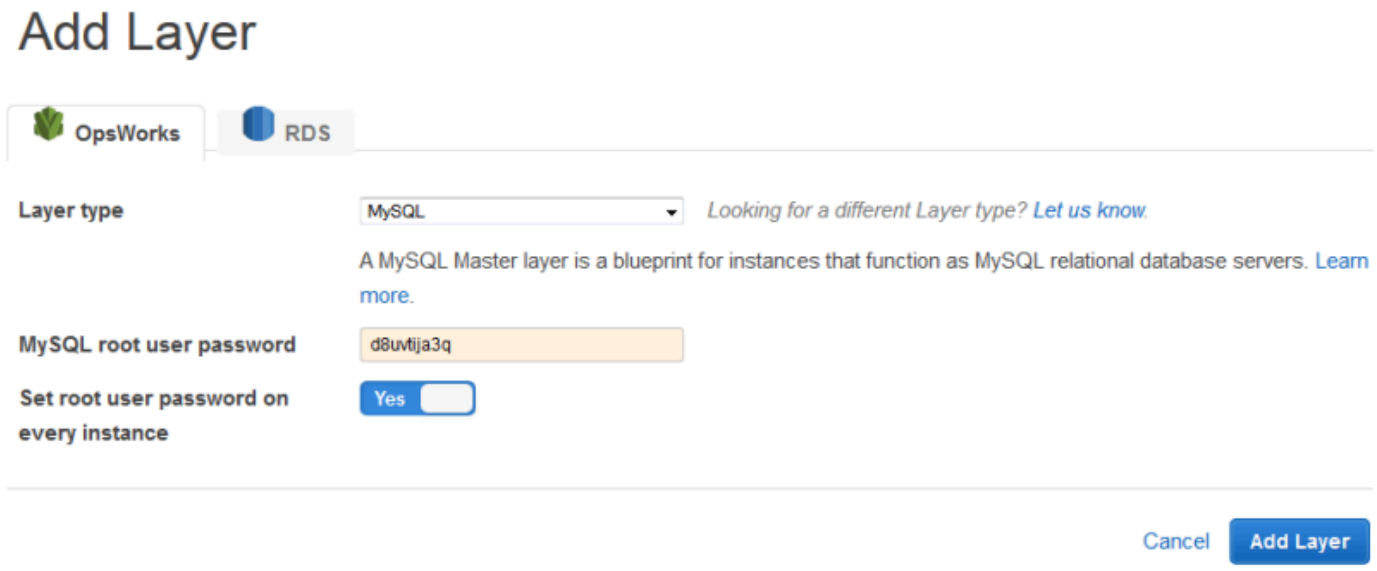
- Lapisan AWS OpsWorks MySQL Stacks adalah cetak biru untuk membuat EC2 instance Amazon yang menjadi tuan rumah master database MySQL.
- Lapisan layanan Amazon RDS menyediakan cara untuk menggabungkan instance Amazon RDS ke dalam tumpukan.

Anda juga dapat menggunakan database lain, seperti Amazon DynamoDB, atau membuat lapisan khusus untuk mendukung database seperti MongoDB. Untuk informasi selengkapnya, lihat the section called “Menggunakan Back-end Data Store”.

Contoh ini menggunakan layer MySQL.

Untuk menambahkan lapisan MySQL MyStack

1. Pada halaman Layers, klik + Layer.
2. Pada halaman Add Layer, untuk tipe Layer, pilih MySQL, terima pengaturan default, dan klik Add Layer.



Untuk menambahkan instance ke lapisan MySQL

1. Pada baris MySQL halaman Layers, klik Add an instance.
2. Pada halaman Instances, di bawah MySQL, klik Tambahkan instance.
3. Terima default dan klik Add instance, tetapi jangan memulainya.

> **Note**
>
> AWS OpsWorks Stacks secara otomatis membuat database bernama menggunakan nama pendek aplikasi, simplephpapp untuk contoh ini. Anda akan memerlukan nama ini jika Anda ingin menggunakan resep Chef untuk berinteraksi dengan database.

Langkah 3.2: Perbarui Sederhana PHPApp

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk memulai, Anda memerlukan versi baru Simple PHPApp yang menggunakan penyimpanan data back-end. Dengan AWS OpsWorks Stacks, mudah untuk memperbarui aplikasi. Jika Anda menggunakan repositori Git atau Subversion, Anda dapat memiliki cabang repositori terpisah untuk setiap versi aplikasi. Contoh aplikasi menyimpan versi aplikasi yang menggunakan database back-end di cabang version2 repositori Git. Anda hanya perlu memperbarui konfigurasi aplikasi untuk menentukan cabang baru dan menerapkan ulang aplikasi.

Untuk memperbarui Sederhana PHPApp

1. Buka Halaman Edit Aplikasi

   Di panel navigasi, klik Aplikasi dan kemudian klik edit di kolom Tindakan PHPApp baris sederhana.

2. Perbarui Konfigurasi Aplikasi

   Ubah pengaturan berikut.

   Cabang/Revisi

   Pengaturan ini menunjukkan cabang repositori aplikasi. Versi pertama Simple PHPApp tidak terhubung ke database. Untuk menggunakan versi aplikasi yang diaktifkan database, tetapkan nilai ini ke. **`version2`**

   Akar dokumen

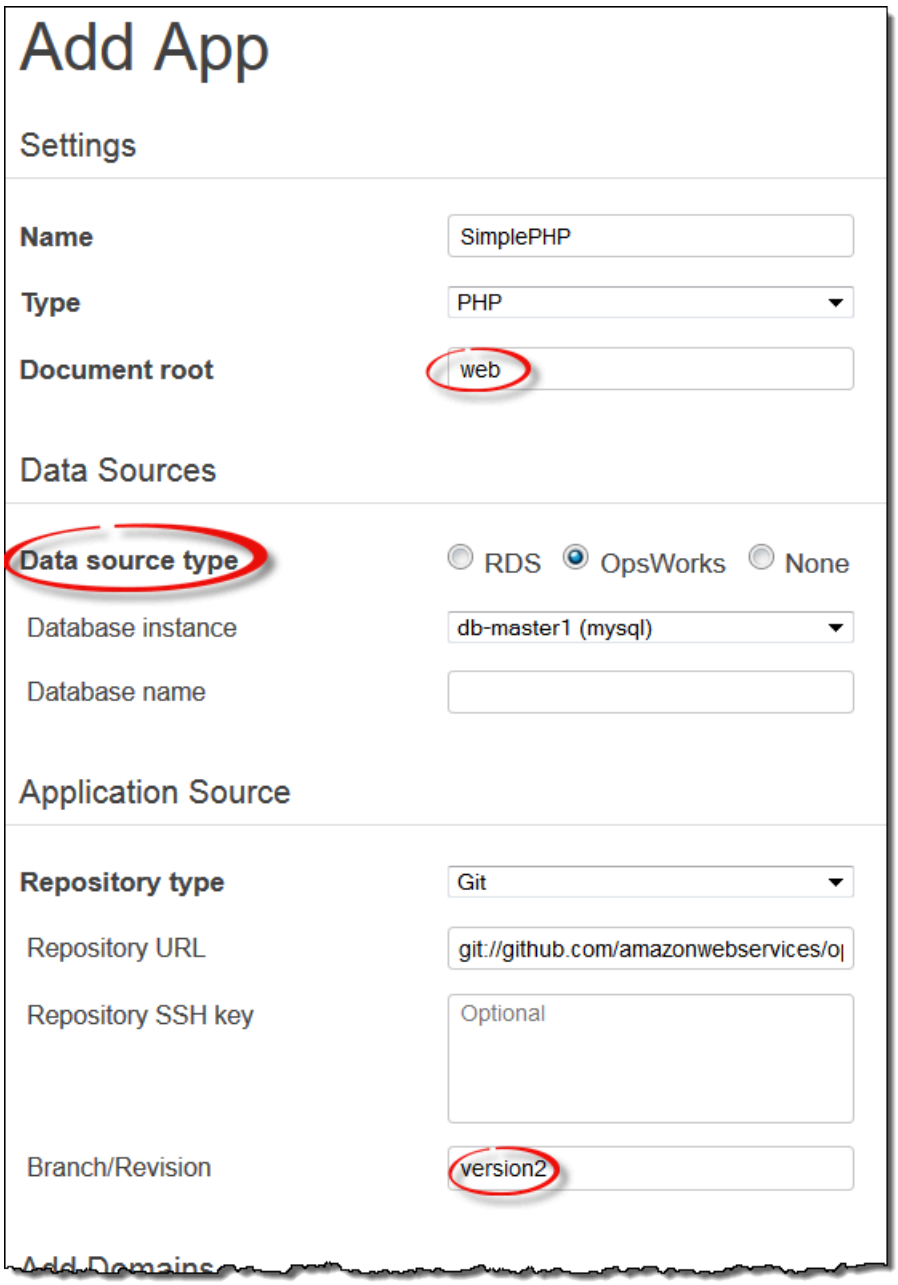
   Setelan ini menentukan folder root aplikasi Anda. Versi pertama Simple PHPApp menggunakan pengaturan default, yang menginstal `index.php` di folder root standar server (`/srv/www`untuk aplikasi PHP). Jika Anda menentukan subfolder di sini—hanya namanya, tidak ada '/' terkemuka —AWS OpsWorks Tumpukan menambahkannya ke jalur folder

standar. Versi 2 dari Simple PHPApp harus masuk`/srv/www/web`, jadi atur root Document ke**`web`**.

Jenis sumber data

Pengaturan ini mengaitkan server database dengan aplikasi. Contoh menggunakan instance MySQL yang Anda buat pada langkah sebelumnya, jadi atur tipe sumber Data OpsWorks ke dan instance Database ke instance yang Anda buat pada langkah sebelumnya, db-master1 (mysql). Biarkan nama Database kosong; AWS OpsWorks Stacks akan membuat database di server bernama dengan nama pendek aplikasi, simplephpapp.

Kemudian klik Simpan untuk menyimpan konfigurasi baru.



3. Mulai contoh MySQL.

Setelah memperbarui aplikasi, AWS OpsWorks Stacks akan secara otomatis menerapkan versi aplikasi baru ke instance server aplikasi baru saat Anda memulainya. Namun, AWS OpsWorks Stacks tidak secara otomatis menerapkan versi aplikasi baru ke instance server yang ada; Anda harus melakukannya secara manual, seperti yang dijelaskan dalam. Langkah 2.4: Membuat dan Menyebarkan Aplikasi - Chef 11 Anda dapat menerapkan Simple yang diperbarui PHPApp sekarang, tetapi untuk contoh ini, lebih baik menunggu sebentar.

Penyimpangan Singkat: Buku Masak, Resep, dan Atribut Tumpukan AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda sekarang memiliki server aplikasi dan database, tetapi mereka tidak cukup siap untuk digunakan. Anda masih perlu menyiapkan database dan mengonfigurasi pengaturan koneksi aplikasi. AWS OpsWorks Stacks tidak menangani tugas-tugas ini secara otomatis, tetapi mendukung buku masak Chef, resep, dan atribut dinamis. Anda dapat menerapkan sepasang resep, satu untuk menyiapkan database dan satu lagi untuk mengonfigurasi pengaturan koneksi aplikasi, dan meminta AWS OpsWorks Stacks menjalankannya untuk Anda.

Buku masak phpapp, yang berisi resep yang diperlukan, sudah diterapkan dan siap digunakan; Anda bisa langsung melompat ke Langkah 3.3: Tambahkan Buku Masak Kustom ke MyStack jika Anda mau. Jika Anda ingin tahu lebih banyak, bagian ini memberikan beberapa latar belakang tentang buku masak dan resep dan menjelaskan cara kerja resep. Untuk melihat buku masak itu sendiri, buka buku masak phpapp.

Topik

- Resep dan Atribut
- Mengatur Database
- Connect Aplikasi ke Database

Resep dan Atribut

Resep Chef pada dasarnya adalah aplikasi Ruby khusus yang melakukan tugas-tugas pada instance seperti menginstal paket, membuat file konfigurasi, menjalankan perintah shell, dan sebagainya. Kelompok resep terkait diatur ke dalam buku masak, yang juga berisi file pendukung seperti templat untuk membuat file konfigurasi.

AWS OpsWorks Stacks memiliki satu set buku masak yang mendukung lapisan bawaan. Anda juga dapat membuat buku masak khusus dengan resep Anda sendiri untuk melakukan tugas khusus

pada instance Anda. Topik ini memberikan pengantar singkat tentang resep dan menunjukkan cara menggunakannya untuk menyiapkan database dan mengonfigurasi setelan koneksi aplikasi. Untuk informasi lebih lanjut tentang buku masak dan resep, lihat Buku Masak dan Resep atauMenyesuaikan Tumpukan AWS OpsWorks.

Resep biasanya bergantung pada atribut Chef untuk data input:

- Beberapa atribut ini didefinisikan oleh Chef dan memberikan informasi dasar tentang instance seperti sistem operasi.
- AWS OpsWorks Stacks mendefinisikan sekumpulan atribut yang berisi informasi tentang tumpukan—seperti konfigurasi lapisan—dan tentang aplikasi yang digunakan—seperti repositori aplikasi.

  Anda dapat menambahkan atribut kustom ke set ini dengan menetapkan JSON kustom ke tumpukan atau penerapan.
- Buku masak Anda juga dapat menentukan atribut, yang khusus untuk buku masak.

  Atribut buku masak phpapp didefinisikan dalam. `attributes/default.rb`

Untuk daftar lengkap atribut AWS OpsWorks Stacks, lihat Konfigurasi Stack dan Atribut Deployment: Linux danAtribut Buku Masak Bawaan. Untuk informasi selengkapnya, lihat Mengesampingkan Atribut.

Atribut diatur dalam struktur hierarkis, yang dapat direpresentasikan sebagai objek JSON.

Anda memasukkan data ini ke dalam aplikasi Anda dengan menggunakan sintaks simpul Chef, seperti berikut ini:

```
[:deploy][:simplephpapp][:database][:username]
```

`deploy`Node ini memiliki satu node aplikasi,`simplephpapp`, yang berisi informasi tentang database aplikasi, repositori Git, dan sebagainya. Contoh ini mewakili nilai nama pengguna database, yang memutuskan untuk. `root`

Mengatur Database

Resep Pengaturan bawaan lapisan MySQL secara otomatis membuat database untuk aplikasi yang diberi nama dengan nama pendek aplikasi, jadi untuk contoh ini Anda sudah memiliki database bernama simplephpapp. Namun, Anda harus menyelesaikan pengaturan dengan membuat tabel untuk aplikasi untuk menyimpan datanya. Anda dapat membuat tabel secara manual, tetapi

pendekatan yang lebih baik adalah menerapkan resep khusus untuk menangani tugas, dan meminta AWS OpsWorks Stacks menjalankannya untuk Anda. Bagian ini menjelaskan bagaimana resep,`dbsetup.rb`, diimplementasikan. Prosedur untuk menjalankan resep AWS OpsWorks Stacks dijelaskan nanti.

Untuk melihat resep di repositori, buka dbsetup.rb. Contoh berikut menunjukkan `dbsetup.rb` kode.

`execute`adalah sumber daya Chef yang menjalankan perintah tertentu. Dalam hal ini, itu adalah perintah MySQL yang membuat tabel. `not_if`Arahan memastikan bahwa perintah tidak berjalan jika tabel yang ditentukan sudah ada. Untuk informasi selengkapnya tentang sumber daya Chef, lihat Tentang Sumber Daya dan Penyedia.

Resep menyisipkan nilai atribut ke dalam string perintah, menggunakan sintaks node dibahas sebelumnya. Misalnya, berikut ini menyisipkan nama pengguna database.

```
#{deploy[:database][:username]}
```

Mari kita bongkar kode yang agak samar ini:

- Untuk setiap iterasi, `deploy` disetel ke node aplikasi saat ini, sehingga diselesaikan menjadi. `[:deploy][:app_name]` Untuk contoh ini, ia memutuskan untuk. `[:deploy][:simplephpapp]`
- Menggunakan nilai atribut deployment yang ditunjukkan sebelumnya, seluruh node menyelesaikan. `root`
- Anda membungkus node di # {} untuk memasukkannya ke dalam string.

Sebagian besar node lain menyelesaikan dengan cara yang sama. Pengecualiannya adalah`#{node[:phpapp][:dbtable]}`, yang didefinisikan oleh file atribut buku masak kustom dan diselesaikan ke nama tabel,. `urler` Oleh karena itu, perintah aktual yang berjalan pada instance MySQL adalah:

```
"/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))'
"
```

Perintah ini membuat tabel bernama `urler` dengan id, author, dan bidang pesan, menggunakan kredensyal dan nama database dari atribut deployment.

Connect Aplikasi ke Database

Bagian kedua dari teka-teki adalah aplikasi, yang membutuhkan informasi koneksi seperti kata sandi database untuk mengakses tabel. Sederhana PHPApp secara efektif hanya memiliki satu file yang berfungsi,`app.php`; semua yang `index.php` dilakukan adalah memuat`app.php`.

`app.php`termasuk`db-connect.php`, yang menangani koneksi database, tetapi file itu tidak ada dalam repositori. Anda tidak dapat membuat `db-connect.php` terlebih dahulu karena mendefinisikan database berdasarkan instance tertentu. Sebagai gantinya, `appsetup.rb` resep menghasilkan `db-connect.php` menggunakan data koneksi dari atribut penerapan.

Untuk melihat resep di repositori, buka appsetup.rb. Contoh berikut menunjukkan `appsetup.rb` kode.

Seperti`dbsetup.rb`, `appsetup.rb` mengulangi aplikasi di `deploy` simpul—hanya simplephpapp lagi—. Ini menjalankan blok kode dengan `script` sumber daya dan `template` sumber daya.

Sumber `script` daya menginstal Composer —manajer ketergantungan untuk aplikasi PHP. Kemudian menjalankan `install` perintah Composer untuk menginstal dependensi untuk aplikasi sampel ke direktori root aplikasi.

`template`Sumber daya menghasilkan `db-connect.php` dan memasukkannya`/srv/www/simplephpapp/current`. Perhatikan hal berikut:

- Resep menggunakan pernyataan bersyarat untuk menentukan pemilik file, yang tergantung pada sistem operasi instance.
- `only_if`Arahan memberitahu Chef untuk menghasilkan template hanya jika direktori yang ditentukan ada.

`template`Sumber daya beroperasi pada template yang pada dasarnya memiliki konten dan struktur yang sama dengan file terkait tetapi menyertakan placeholder untuk berbagai nilai data. `source`Parameter menentukan template,`db-connect.php.erb`, yang ada di `templates/default` direktori buku masak phpapp, dan berisi yang berikut:

Ketika Chef memproses template, itu menggantikan <%= => placeholder dengan nilai variabel yang sesuai dalam sumber daya template, yang pada gilirannya diambil dari atribut deployment. Oleh karena itu, file yang dihasilkan adalah:

Langkah 3.3: Tambahkan Buku Masak Kustom ke MyStack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda menyimpan buku masak khusus di repositori, seperti aplikasi. Setiap tumpukan dapat memiliki repositori yang berisi satu set buku masak khusus. Anda kemudian mengarahkan AWS OpsWorks Stacks untuk menginstal buku masak khusus Anda pada instance tumpukan.

1. Klik Stack di panel navigasi untuk melihat halaman untuk tumpukan saat ini.
2. Klik Pengaturan Stack, lalu klik Edit.
3. Ubah konfigurasi tumpukan sebagai berikut:
   - Gunakan Buku Masak Koki khusus - Ya
   - Jenis repositori - Git
   - URL Repositori — **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`**
4. Klik Simpan untuk memperbarui konfigurasi tumpukan.



AWS OpsWorks Stacks kemudian menginstal isi repositori buku masak Anda di semua instance tumpukan. Jika Anda membuat instance baru, AWS OpsWorks Stacks secara otomatis menginstal repositori buku masak.

> **Note**
>
> Jika Anda perlu memperbarui salah satu buku masak Anda, atau menambahkan buku masak baru ke repositori, Anda dapat melakukannya tanpa menyentuh pengaturan tumpukan. AWS OpsWorks Stacks akan secara otomatis menginstal buku masak yang diperbarui pada semua instance baru. Namun, AWS OpsWorks Stacks tidak secara otomatis menginstal buku masak yang diperbarui pada instance online stack. Anda harus secara eksplisit mengarahkan AWS OpsWorks Stacks untuk memperbarui buku masak dengan menjalankan perintah stack. `Update Cookbooks` Untuk informasi selengkapnya, lihat Perintah Tumpukan.

Langkah 3.4: Jalankan Resep

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda memiliki buku masak khusus Anda, Anda perlu menjalankan resep pada contoh yang sesuai. Anda bisa menjalankannya secara manual. Namun, resep biasanya harus dijalankan pada titik yang dapat diprediksi dalam siklus hidup instance, seperti setelah instance boot atau saat Anda menerapkan aplikasi. Bagian ini menjelaskan pendekatan yang jauh lebih sederhana: minta AWS OpsWorks Stacks menjalankannya secara otomatis untuk Anda pada waktu yang tepat.

AWS OpsWorks Stacks mendukung serangkaian peristiwa siklus hidup yang menyederhanakan menjalankan resep. Misalnya, peristiwa Penyetelan terjadi setelah instance selesai booting dan peristiwa Deploy terjadi saat Anda menerapkan aplikasi. Setiap lapisan memiliki satu set resep bawaan yang terkait dengan setiap peristiwa siklus hidup. Ketika peristiwa siklus hidup terjadi pada sebuah instance, agen menjalankan resep terkait untuk setiap lapisan instance. Agar AWS

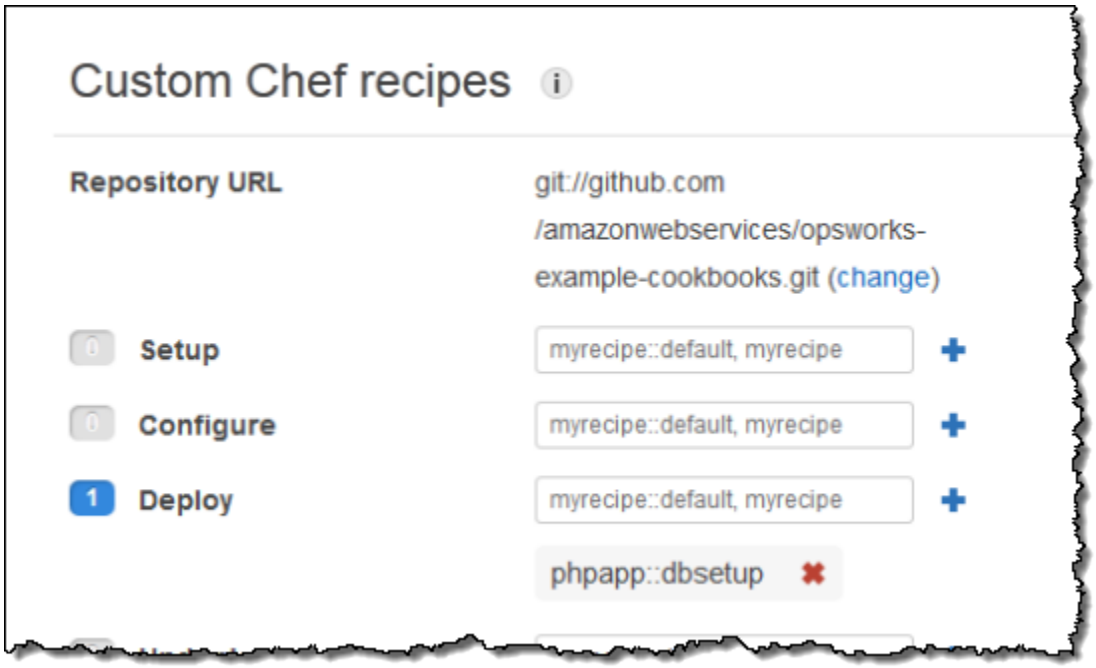
OpsWorks Stacks menjalankan resep khusus secara otomatis, tambahkan ke acara siklus hidup yang sesuai pada lapisan yang sesuai dan agen akan menjalankan resep setelah resep bawaan selesai.

Untuk contoh ini, Anda perlu menjalankan dua resep, `dbsetup.rb` pada contoh My SQLinstance dan `appsetup.rb` pada PHP App Server.

> **Note**
>
> Anda menentukan resep di konsol dengan menggunakan *`recipe_name`* format*`cookbook_name`*::, di mana *`recipe_name`* tidak termasuk ekstensi.rb. Misalnya, Anda merujuk `dbsetup.rb` sebagai**`phpapp::dbsetup`**.

Untuk menetapkan resep kustom ke peristiwa siklus hidup

1. Pada halaman Layers, untuk MySQL, klik Resep dan kemudian klik Edit.
2. Di bagian Resep Koki Kustom, masukkan **`phpapp::dbsetup`**untuk Menerapkan.



3. Klik ikon + untuk menetapkan resep ke acara dan klik Simpan untuk menyimpan konfigurasi layer baru.
4. Kembali ke halaman Layers dan ulangi prosedur untuk menetapkan **`phpapp::appsetup`** ke event Deploy layer PHP App Server.

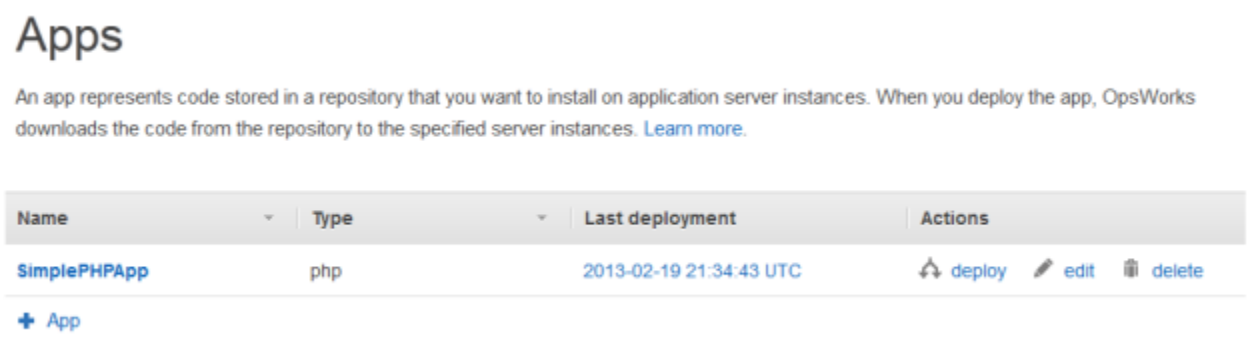
Langkah 3.5: Menyebarkan SederhanaPHPApp, Versi 2

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

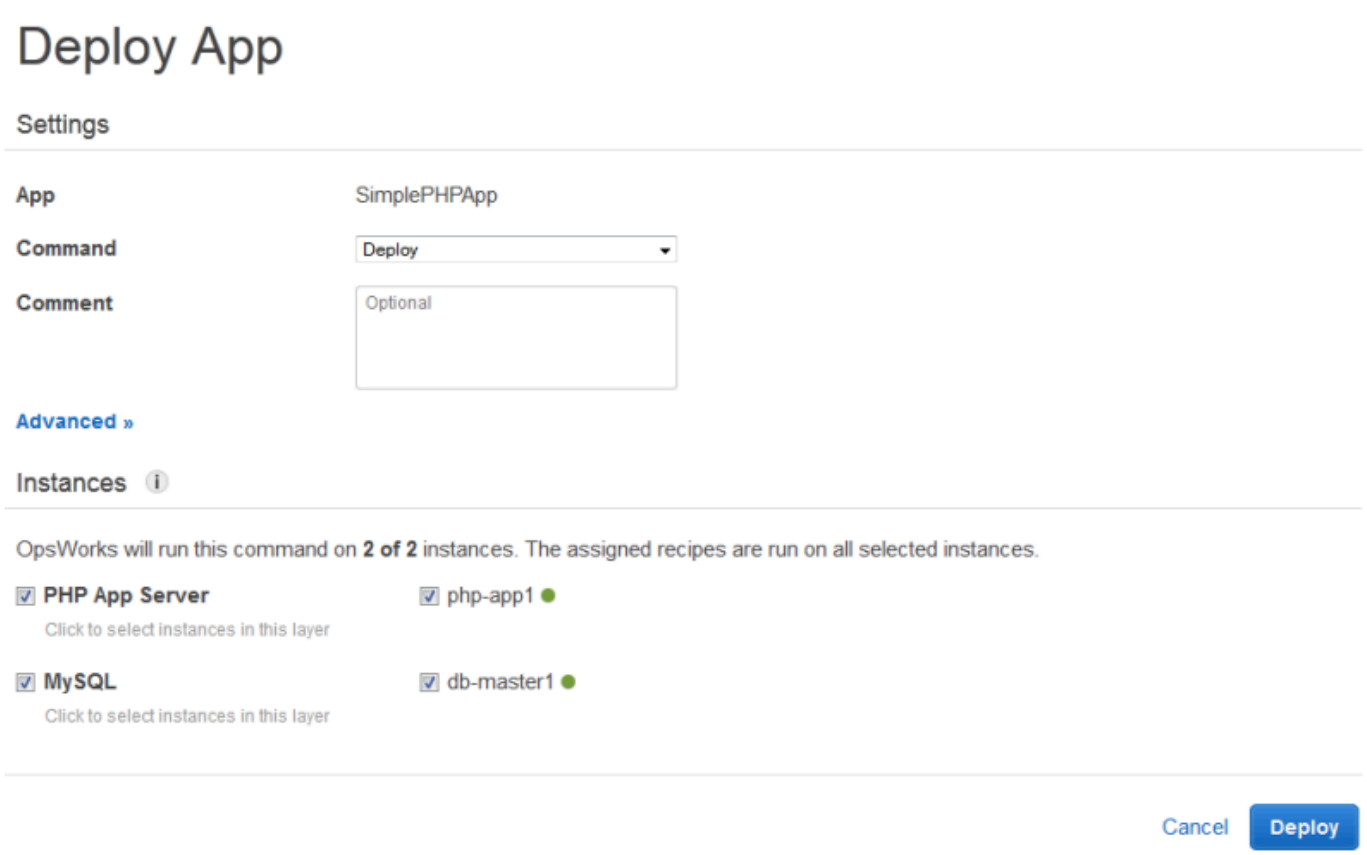
Langkah terakhir adalah menerapkan versi baru SimplePHPApp.

Untuk menyebarkan Sederhana PHPApp

1. Di halaman Aplikasi, klik terapkan di Tindakan PHPApp aplikasi Sederhana.



2. Terima default dan klik Deploy.



Saat Anda mengklik Deploy di halaman Deploy App, Anda akan memicu peristiwa siklus hidup Deploy, yang memberi tahu agen untuk menjalankan resep Deploy mereka. Secara default, Anda memicu peristiwa pada semua instance tumpukan. Resep Deploy bawaan menerapkan aplikasi hanya ke instance yang sesuai untuk jenis aplikasi, instance PHP App Server dalam kasus ini. Namun, seringkali berguna untuk memicu peristiwa Deploy pada instance lain, untuk memungkinkan mereka merespons penerapan aplikasi. Dalam hal ini, Anda juga ingin memicu Deploy pada instance MySQL untuk mengatur database.

Perhatikan hal berikut:

- Agen pada instance PHP App Server menjalankan resep bawaan layer, diikuti oleh`appsetup.rb`, yang mengonfigurasi koneksi database aplikasi.
- Agen pada instance MySQL tidak menginstal apa pun, tetapi `dbsetup.rb` berjalan untuk membuat tabel urler.

Ketika penyebaran selesai, Status akan berubah menjadi sukses di halaman Deployment.

Langkah 3.6: Jalankan Sederhana PHPApp

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah status penerapan berubah menjadi sukses, Anda dapat menjalankan PHPApp versi Simple baru, sebagai berikut.

Untuk menjalankan Sederhana PHPApp

1. Pada halaman Instances, klik alamat IP publik di baris php-app1.

   Anda akan melihat halaman berikut di browser Anda.



2. Klik Bagikan Pemikiran Anda dan ketikkan sesuatu seperti **`Hello world!`** untuk Pikiran Anda dan nama Anda untuk Nama Anda. Kemudian klik Kirim Pikiran Anda untuk menambahkan pesan ke database.

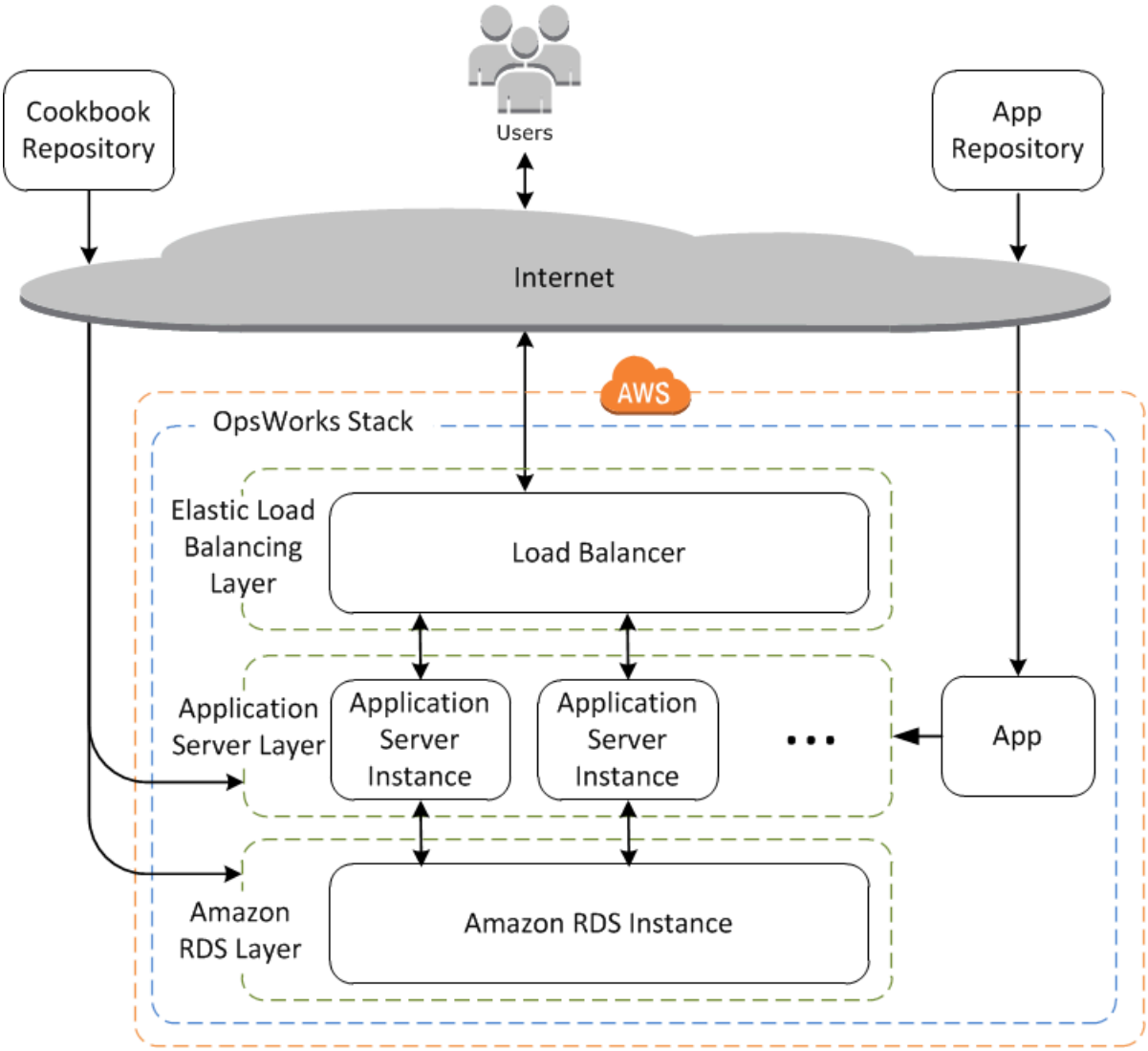


3. Klik Kembali untuk melihat semua pesan dalam database.

Langkah 4: Scale Out MyStack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

MyStack Saat ini hanya memiliki satu server aplikasi. Tumpukan produksi mungkin akan membutuhkan beberapa server aplikasi untuk menangani lalu lintas masuk dan penyeimbang beban untuk mendistribusikan lalu lintas masuk secara merata di seluruh server aplikasi. Arsitekturnya akan terlihat seperti berikut:



AWS OpsWorks Tumpukan membuatnya mudah untuk skala tumpukan. Bagian ini menjelaskan dasar-dasar cara menskalakan tumpukan dengan menambahkan instance Server Aplikasi PHP 24/7 kedua ke MyStack dan menempatkan kedua instance di belakang penyeimbang beban Elastic Load Balancing. Anda dapat dengan mudah memperpanjang prosedur untuk menambahkan jumlah instance 24/7 yang sewenang-wenang, atau Anda dapat menggunakan instance berbasis waktu atau berbasis beban agar Stacks menskalakan tumpukan Anda AWS OpsWorks secara otomatis. Untuk informasi selengkapnya, lihat Mengelola beban dengan instans berbasis waktu dan berbasis beban.

Langkah 4.1: Tambahkan Load Balancer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Elastic Load Balancing adalah layanan AWS yang secara otomatis mendistribusikan lalu lintas aplikasi masuk di beberapa instans Amazon. EC2 Selain mendistribusikan lalu lintas, Elastic Load Balancing melakukan hal berikut:

- Mendeteksi EC2 contoh Amazon yang tidak sehat.

  Ini mengalihkan lalu lintas ke contoh sehat yang tersisa sampai kasus yang tidak sehat dipulihkan.
- Secara otomatis menskalakan kapasitas penanganan permintaan sebagai respons terhadap lalu lintas yang masuk

> **Note**
>
> Penyeimbang beban dapat melayani dua tujuan. Yang jelas adalah menyamakan beban pada server aplikasi Anda. Selain itu, banyak situs lebih suka mengisolasi server aplikasi dan database mereka dari akses pengguna langsung. Dengan AWS OpsWorks Stacks, Anda dapat melakukan ini dengan menjalankan tumpukan Anda di cloud pribadi virtual (VPC) dengan subnet publik dan pribadi, sebagai berikut.
>
> - Letakkan server aplikasi dan database di subnet pribadi, di mana mereka dapat diakses oleh instance lain di VPC tetapi tidak oleh pengguna.
> - Mengarahkan lalu lintas pengguna ke penyeimbang beban di subnet publik, yang kemudian meneruskan lalu lintas ke server aplikasi di subnet pribadi dan mengembalikan tanggapan kepada pengguna.
>
> Untuk informasi selengkapnya, lihat Menjalankan Stack di VPC. Untuk AWS CloudFormation template yang memperluas contoh dalam panduan ini agar berjalan di VPC, unduh file tersebut. `OpsWorksVPCtemplates.zip`

Meskipun Elastic Load Balancing sering disebut sebagai lapisan, ia bekerja sedikit berbeda dari lapisan bawaan lainnya. Alih-alih membuat layer dan menambahkan instance ke dalamnya, Anda membuat penyeimbang beban Elastic Load Balancing dengan menggunakan konsol EC2 Amazon

dan kemudian melampirkannya ke salah satu layer yang ada, biasanya lapisan server aplikasi. AWS OpsWorks Stacks kemudian mendaftarkan instance layer yang ada dengan layanan dan secara otomatis menambahkan instance baru. Prosedur berikut menjelaskan cara menambahkan penyeimbang beban ke MyStack lapisan PHP App Server.

> **Note**
>
> AWS OpsWorks Stacks tidak mendukung Application Load Balancer. Anda hanya dapat menggunakan Classic Load Balancer dengan AWS OpsWorks Stacks.

Untuk melampirkan load balancer ke lapisan PHP App Server

1. Gunakan EC2 konsol Amazon untuk membuat penyeimbang beban baru. MyStack Detailnya tergantung pada apakah akun Anda mendukung EC2 Classic. Untuk informasi selengkapnya, lihat Memulai Elastic Load Balancing. Saat Anda menjalankan wizard Create Load Balancer, konfigurasikan penyeimbang beban sebagai berikut:

   Tentukan Load Balancer

   Tetapkan penyeimbang beban nama yang mudah dikenali, seperti PHP-LB, untuk membuatnya lebih mudah ditemukan di konsol Stacks. AWS OpsWorks Kemudian pilih Lanjutkan untuk menerima default untuk pengaturan yang tersisa.

   Jika Anda memilih VPC dengan satu atau lebih subnet dari menu Create LB Inside, Anda harus memilih subnet untuk setiap zona ketersediaan tempat Anda ingin lalu lintas dirutekan oleh penyeimbang beban Anda.

   Tetapkan Grup Keamanan

   Jika akun Anda mendukung VPC default, wizard akan menampilkan halaman ini untuk menentukan grup keamanan penyeimbang beban. Itu tidak menampilkan halaman ini untuk EC2 Klasik.

   Untuk panduan ini, pilih grup keamanan VPC default.

   Konfigurasikan Pengaturan Keamanan

   Jika Anda memilih HTTPS sebagai Protokol Load Balancer pada halaman Define Load Balancer, konfigurasikan setelan protokol sertifikat, sandi, dan SSL di halaman ini. Untuk panduan ini, terima default, dan pilih Konfigurasi Pemeriksaan Kesehatan.

Konfigurasikan Pemeriksaan Kesehatan

Atur jalur ping ke **/** dan terima default untuk pengaturan yang tersisa.

Tambahkan EC2 Instans

Pilih Lanjutkan; AWS OpsWorks Tumpukan secara otomatis mendaftarkan instance dengan penyeimbang beban.

Tambahkan Tag

Tambahkan tag untuk membantu Anda menemukan. Setiap tag adalah pasangan kunci dan nilai; misalnya, Anda dapat menentukan **`Description`** sebagai kunci dan **`Test LB`** sebagai nilai untuk tujuan penelusuran.

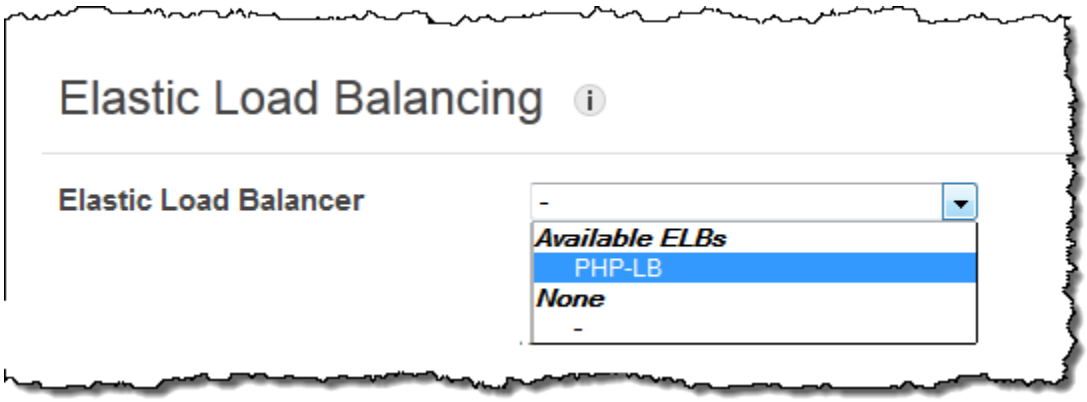
Ulasan

Tinjau pilihan Anda, pilih Buat, lalu pilih Tutup, yang memulai penyeimbang beban.

2. Jika akun Anda mendukung VPC default, setelah Anda memulai penyeimbang beban, Anda harus memastikan bahwa grup keamanannya memiliki aturan masuk yang sesuai. Aturan default tidak menerima lalu lintas masuk apa pun.

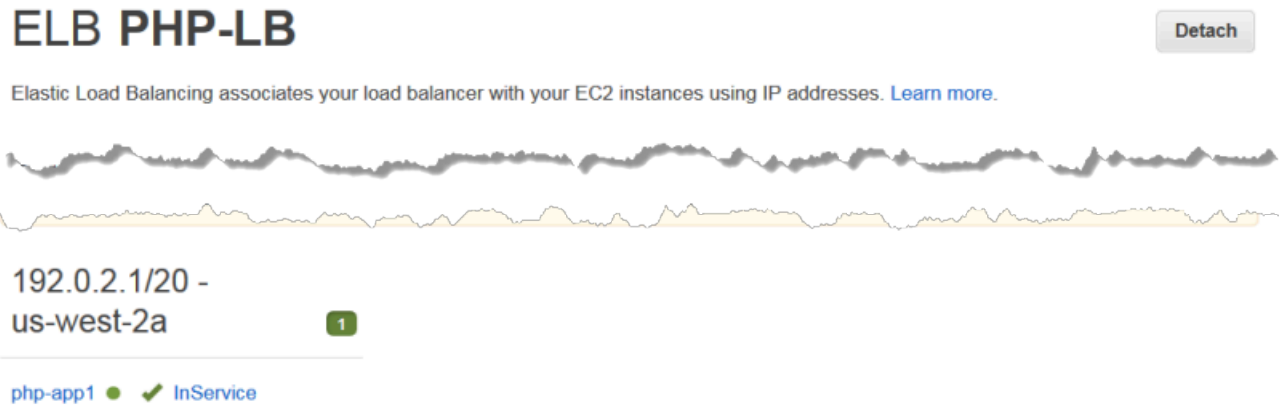
   1. Pilih Grup Keamanan di panel EC2 navigasi Amazon.
   2. Pilih grup keamanan VPC default
   3. Pilih Edit pada tab Inbound.
   4. Untuk panduan ini, atur Source to Anywhere, yang mengarahkan penyeimbang beban untuk menerima lalu lintas masuk dari alamat IP apa pun.

3. Kembali ke konsol AWS OpsWorks Stacks. Pada halaman Layers, pilih link Network layer, dan kemudian pilih Edit.
4. Di bawah Elastic Load Balancing, pilih load balancer yang Anda buat di Langkah 1, lalu pilih Simpan.



Setelah Anda melampirkan load balancer ke layer, AWS OpsWorks Stacks secara otomatis mendaftarkan instance layer saat ini, dan menambahkan instance baru saat mereka online.

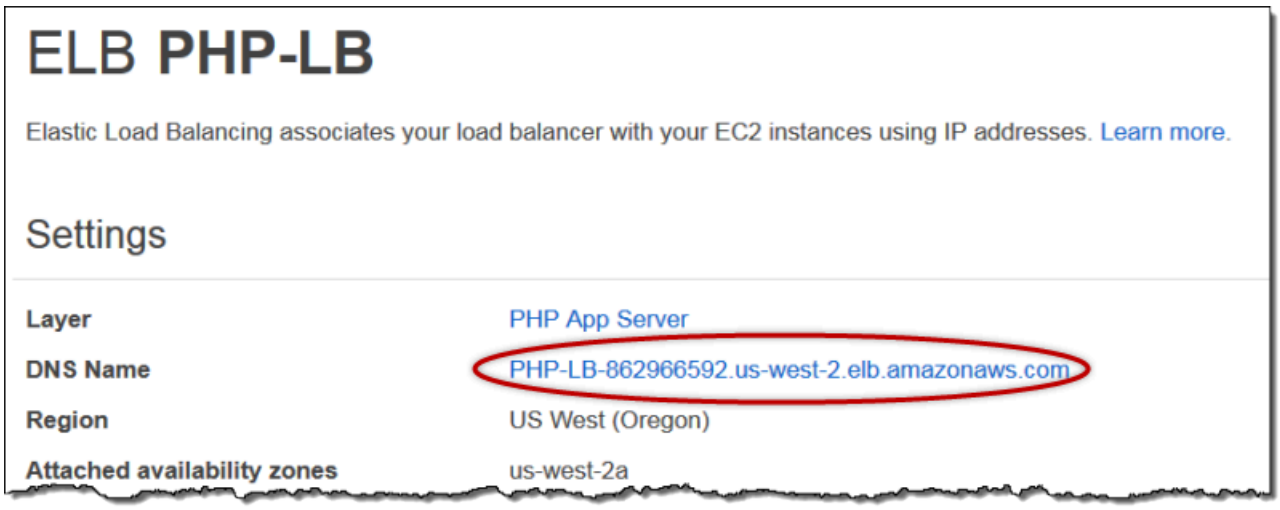
5. Pada halaman Layers, klik nama load balancer untuk membuka halaman detailnya. Ketika pendaftaran selesai dan instance melewati pemeriksaan kesehatan, AWS OpsWorks Stacks menunjukkan tanda centang hijau di sebelah instance pada halaman penyeimbang beban.



Anda sekarang dapat menjalankan Simple PHPApp dengan mengirimkan permintaan ke load balancer.

Untuk menjalankan Simple PHPApp melalui load balancer

1. Buka halaman detail load balancer lagi, jika belum terbuka.
2. Pada halaman properti, verifikasi status pemeriksaan kesehatan instans dan klik nama DNS load balancer untuk menjalankan Simple. PHPApp Load balancer meneruskan permintaan ke instance PHP App Server dan mengembalikan respons, yang akan terlihat persis sama dengan respons yang Anda dapatkan ketika Anda mengklik alamat IP publik instance PHP App Server.



> **Note**
>
> AWS OpsWorks Stacks juga mendukung penyeimbang HAProxy beban, yang mungkin memiliki keuntungan untuk beberapa aplikasi. Untuk informasi selengkapnya, lihat HAProxy AWS OpsWorks Lapisan Tumpukan.

Langkah 4.2: Tambahkan Instans Server Aplikasi PHP

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang load balancer sudah ada, Anda dapat menskalakan tumpukan dengan menambahkan lebih banyak instance ke lapisan PHP App Server. Dari sudut pandang Anda, operasinya mulus. Setiap kali instance PHP App Server baru online, AWS OpsWorks Stacks secara otomatis mendaftarkannya dengan penyeimbang beban dan menyebarkan SimplePHPApp, sehingga server dapat segera mulai menangani lalu lintas masuk. Untuk singkatnya, topik ini menunjukkan cara menambahkan satu instance PHP App Server tambahan, tetapi Anda dapat menggunakan pendekatan yang sama untuk menambahkan sebanyak yang Anda butuhkan.

Untuk menambahkan instance lain ke lapisan PHP App Server

1. Pada halaman Instans, klik + Instance di bawah PHP App Server.
2. Terima pengaturan default dan klik Add Instance.
3. Klik mulai untuk memulai instance.

Langkah 4.3: Monitor MyStack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

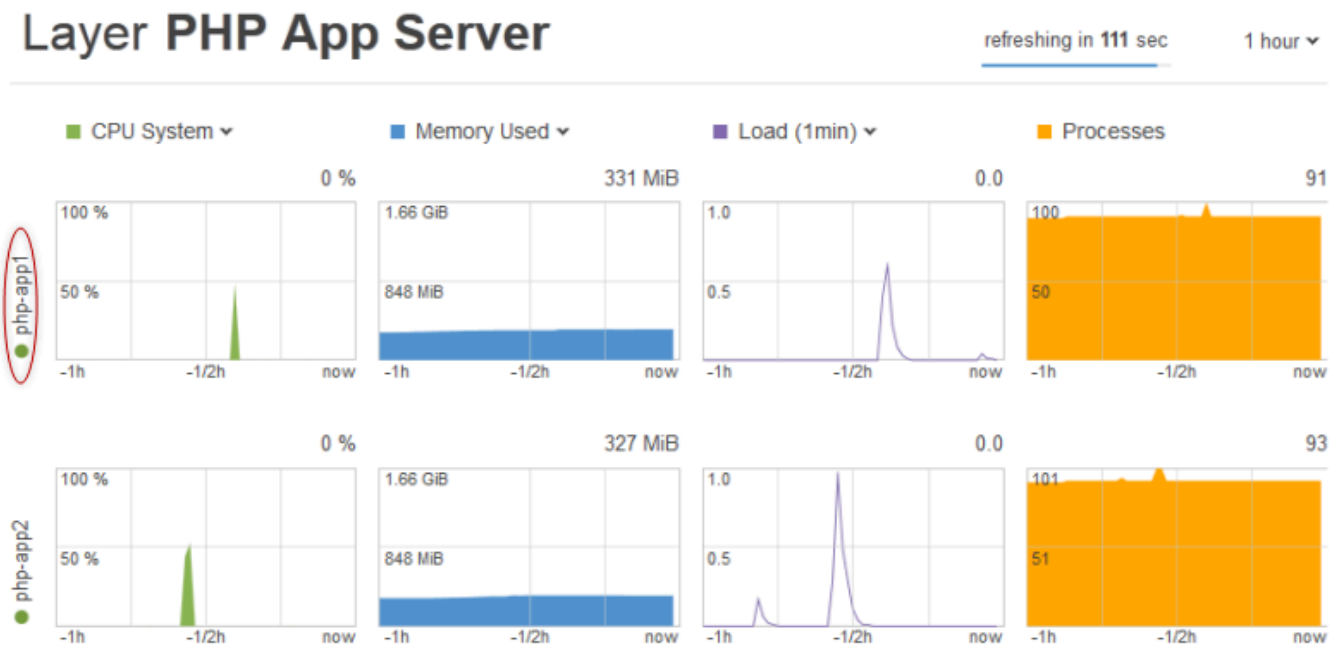
AWS OpsWorks Stacks menggunakan Amazon CloudWatch untuk menyediakan metrik untuk tumpukan dan meringkasnya untuk kenyamanan Anda di halaman Pemantauan. Anda dapat melihat metrik untuk seluruh tumpukan, lapisan tertentu, atau instance tertentu.

Untuk memantau MyStack

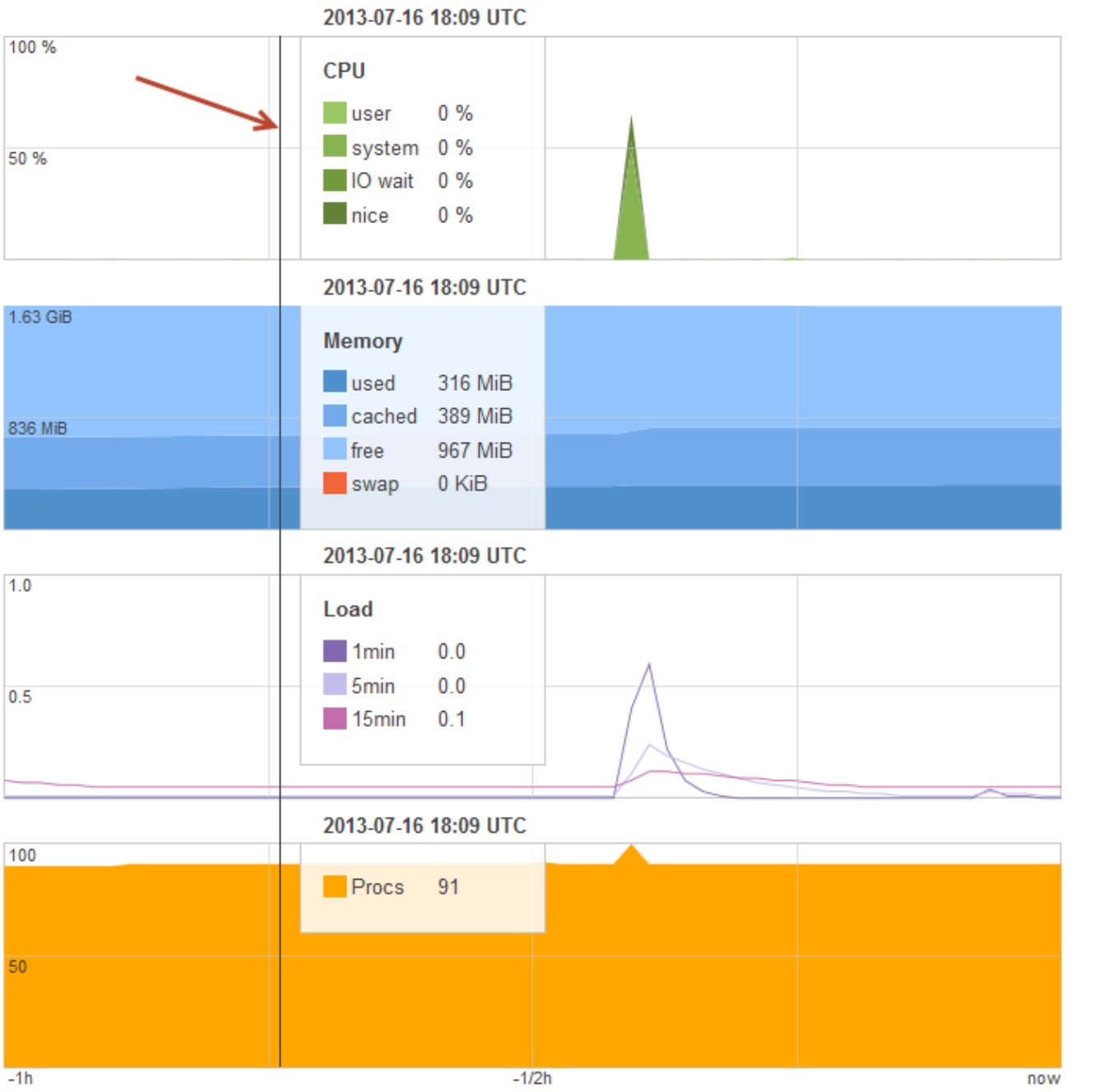
1. Di panel navigasi, klik Monitoring, yang menampilkan satu set grafik dengan metrik rata-rata untuk setiap lapisan. Anda dapat menggunakan menu untuk CPU System, Memory Used, dan Load untuk menampilkan metrik terkait yang berbeda.



2. Klik PHP App Server untuk melihat metrik untuk setiap instance layer.



3. Klik php-app1 untuk melihat metrik untuk contoh itu. Anda dapat melihat metrik untuk titik waktu tertentu dengan menggerakkan slider.


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 juga mendukung server pemantauan Ganglia, yang mungkin memiliki keuntungan untuk beberapa aplikasi. Untuk informasi selengkapnya, lihat Lapisan Ganglia.

Langkah 5: Hapus MyStack

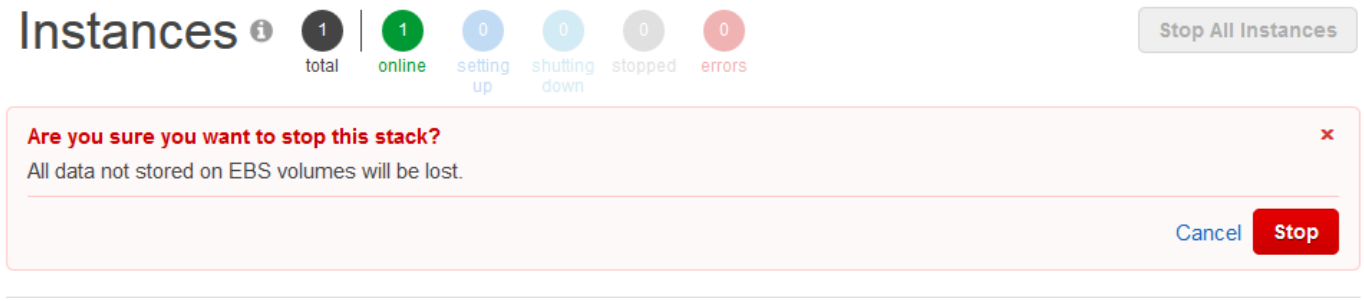
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Segera setelah Anda mulai menggunakan sumber daya AWS seperti EC2 instans Amazon, Anda akan dikenakan biaya berdasarkan penggunaan Anda. Jika Anda selesai untuk saat ini, Anda harus menghentikan kasus sehingga Anda tidak dikenakan biaya yang tidak diinginkan. Jika Anda tidak membutuhkan tumpukan lagi, Anda dapat menghapusnya.

Untuk menghapus MyStack

1. Hentikan semua Instance

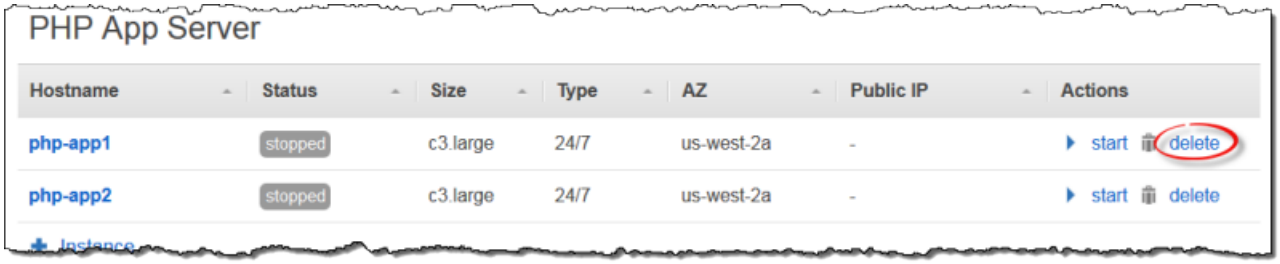
   Pada halaman Instans, klik Hentikan Semua Instans dan klik Berhenti saat diminta konfirmasi operasi.



   Setelah Anda mengklik Stop, AWS OpsWorks Stacks menghentikan EC2 instans Amazon terkait, tetapi tidak semua sumber daya terkait seperti alamat IP Elastis atau volume Amazon EBS.

2. Hapus semua Instance

   Menghentikan instance hanya akan menghentikan EC2 instans Amazon terkait. Setelah status instance dalam keadaan berhenti, Anda harus menghapus setiap instance. Di lapisan PHP App Server klik delete di kolom Actions instance php-app1.

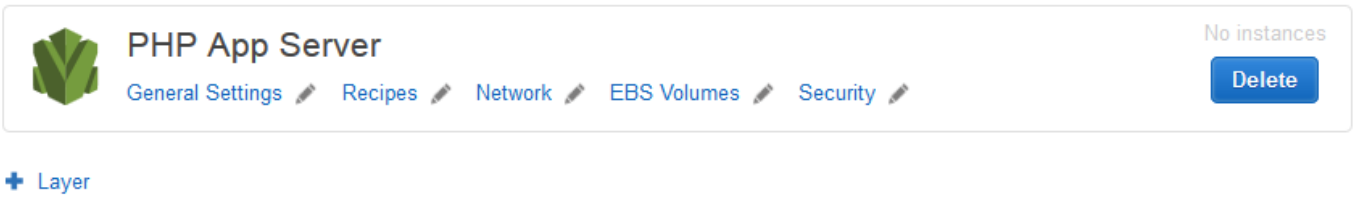


   AWS OpsWorks Stacks kemudian meminta Anda untuk mengonfirmasi penghapusan, dan menunjukkan sumber daya yang bergantung kepada Anda. Anda dapat memilih untuk menyimpan salah satu atau semua sumber daya ini. Contoh ini tidak memiliki sumber daya dependen, jadi cukup klik Hapus.

   Ulangi proses untuk php-app2 dan instance MySQL, db-master1. Perhatikan bahwa db-master1 memiliki volume Amazon Elastic Block Store terkait, yang dipilih secara default. Biarkan dipilih untuk menghapus volume bersama dengan instance.

3. Hapus Layers.

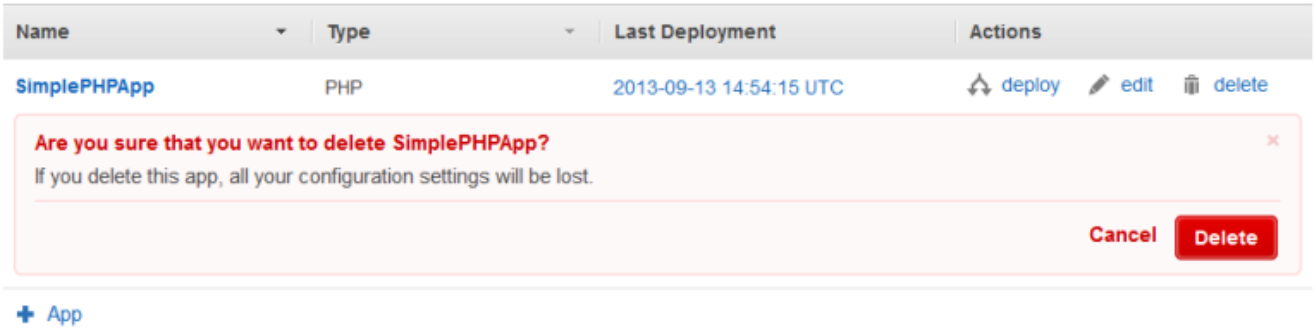
   Pada halaman Layers, klik Delete dan kemudian klik Delete untuk mengonfirmasi.



   Ulangi proses untuk layer MySQL.

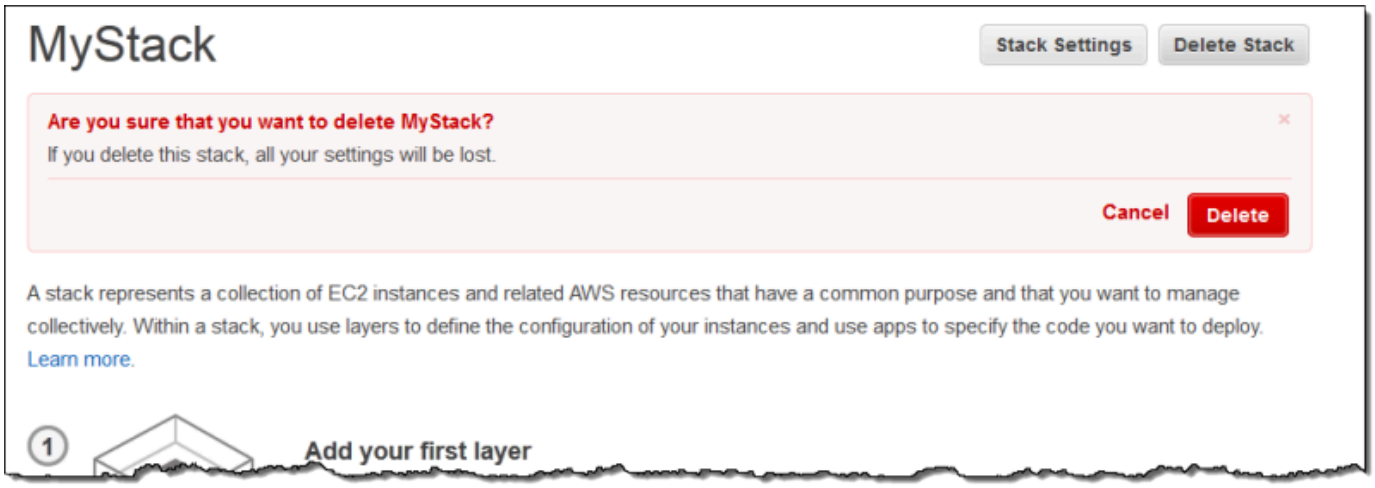
4. Hapus Aplikasi

   Pada halaman Aplikasi, klik hapus di kolom Tindakan PHPApp aplikasi sederhana, lalu klik Hapus untuk mengonfirmasi.



5. Hapus MyStack

   Pada halaman Stack, klik Delete Stack dan kemudian klik Delete untuk mengkonfirmasi.



Anda sekarang telah mencapai akhir dari panduan ini.

Membuat Stack Node.js Pertama Anda

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Contoh ini menjelaskan cara membuat tumpukan Linux yang mendukung server aplikasi Node.js dan cara menyebarkan aplikasi sederhana. Tumpukan terdiri dari komponen-komponen berikut:

- Lapisan Server Aplikasi Node.js dengan dua instance
- Penyeimbang beban Elastic Load Balancing untuk mendistribusikan lalu lintas ke instance server aplikasi
- Lapisan layanan Amazon Relational Database Service (Amazon RDS) yang menyediakan basis data backend

Topik

- Prasyarat
- Menerapkan Aplikasi
- Membuat Server Database dan Load Balancer
- Membuat Stack
- Menyebarkan Aplikasi
- Apa Selanjutnya?

Prasyarat

Panduan ini mengasumsikan hal berikut:

- Anda memiliki akun AWS dan pemahaman dasar tentang cara menggunakan AWS OpsWorks Stacks.

  Jika Anda baru mengenal AWS OpsWorks Stacks atau AWS, pelajari dasar-dasarnya dengan menyelesaikan tutorial pengantar di. Memulai dengan Chef 11 Linux Stacks
- Anda memiliki pemahaman dasar tentang bagaimana menerapkan aplikasi Node.js.

  Jika Anda baru mengenal Node.js, pelajari dasar-dasarnya dengan menyelesaikan tutorial pengantar, seperti Node: Up and Running.
- Anda telah membuat setidaknya satu tumpukan di wilayah AWS yang akan Anda gunakan untuk contoh ini.

  Saat Anda membuat tumpukan pertama di suatu wilayah, AWS OpsWorks Stacks membuat grup keamanan Amazon Elastic Compute Cloud EC2 (Amazon) untuk setiap jenis lapisan. Anda memerlukan grup keamanan ini untuk membuat instance Amazon RDS database (DB). Jika Anda baru mengenal AWS OpsWorks Stacks, kami sarankan Anda menggunakan wilayah yang sama untuk contoh ini yang Anda lakukan ketika Anda mengikuti tutorial diMemulai dengan Chef 11 Linux Stacks. Jika Anda ingin menggunakan wilayah baru, buat tumpukan baru di wilayah tersebut;

tumpukan tidak perlu memiliki lapisan atau instance. Segera setelah Anda membuat tumpukan, AWS OpsWorks Stacks secara otomatis menambahkan satu set grup keamanan ke wilayah tersebut.

- Anda akan membuat tumpukan Anda di VPC default.

  Anda dapat menggunakan EC2 -Classic untuk panduan ini, tetapi beberapa detailnya akan sedikit berbeda. Misalnya, dengan EC2 -Classic, Anda menentukan Availability Zone (AZ) instance, bukan subnetnya.
- Pengguna IAM Anda memiliki izin akses penuh untuk Stacks. AWS OpsWorks

  Untuk alasan keamanan, kami sangat menyarankan agar Anda tidak menggunakan kredensi root akun Anda untuk panduan ini. Sebagai gantinya, buat pengguna dengan izin akses penuh AWS OpsWorks Stacks dan gunakan kredensil tersebut dengan Stacks. AWS OpsWorks Untuk informasi selengkapnya, lihat Membuat Pengguna Administratif.

Menerapkan Aplikasi

Panduan ini menggunakan aplikasi Express sederhana yang terhubung ke instans Amazon RDS DB dan mencantumkan database instans.

Untuk mengimplementasikan aplikasi, buat direktori bernama `nodedb` di lokasi yang nyaman di workstation Anda dan tambahkan tiga file berikut ke dalamnya.

Topik

- Deskriptor Package
- File Tata Letak
- File Kode

Deskriptor Package

Untuk membuat deskriptor paket aplikasi, tambahkan file bernama `package.json` dengan konten berikut ke `nodedb` direktori. `package.json`diperlukan untuk aplikasi Express dan harus ditempatkan di direktori root aplikasi.

```
{
  "name": "Nodejs-DB",
  "description": "Node.js example application",
```

```
  "version": "0.0.1",
  "dependencies": {
    "express": "*",
    "ejs": "*",
    "mysql": "*"
  }
}
```

`package.json`Contoh ini cukup minim. Ini mendefinisikan diperlukan `name` dan `version` atribut dan daftar paket dependen:

- `express`referensi paket Express.
- `ejs`referensi paket EJS, yang digunakan aplikasi untuk menyisipkan teks ke dalam file tata letak HTML.
- `mysql`referensi paket node-mysql, yang digunakan aplikasi untuk terhubung ke instance RDS.

Untuk informasi selengkapnya tentang file deskriptor paket, lihat package.json.

File Tata Letak

Untuk membuat file tata letak aplikasi, tambahkan `views` direktori ke `nodedb` direktori, lalu tambahkan file yang `views` diberi nama `index.html` dengan konten berikut:

```
<!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>
```

Untuk contoh ini, file tata letak adalah dokumen HTML sederhana yang menampilkan beberapa data dari Amazon RDS. Setiap <%= ... => elemen mewakili nilai variabel yang didefinisikan dalam file kode aplikasi, yang kita buat selanjutnya.

File Kode

Untuk membuat file kode aplikasi, tambahkan `server.js` file ke `nodedb` direktori dengan konten berikut.

> **Important**
>
> Dengan AWS OpsWorks Stacks, file kode utama aplikasi Node.js harus diberi nama `server.js` dan ditempatkan di folder root aplikasi.

```
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);
```

Contoh ini menampilkan informasi koneksi database dan juga query server database dan menampilkan database server. Anda dapat dengan mudah menggeneralisasikannya untuk berinteraksi dengan database sesuai kebutuhan. Catatan berikut mengacu pada komentar bernomor dalam kode sebelumnya.

[1] Sertakan data koneksi database

`require`Pernyataan ini mencakup data koneksi database. Seperti yang dijelaskan nanti, saat Anda melampirkan instance database ke aplikasi, AWS OpsWorks Stacks menempatkan data koneksi dalam file bernama`opsworks.js`, yang terlihat mirip dengan berikut ini:

```
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`sada di `shared/config` direktori aplikasi,`/srv/www/app_shortname/shared/config`. Namun, AWS OpsWorks Stacks menempatkan symlink ke `opsworks.js` dalam direktori root aplikasi, sehingga Anda dapat menyertakan objek dengan hanya menggunakan. `require 'opsworks'`

[2] Dapatkan data koneksi database

Kumpulan pernyataan ini menampilkan data koneksi dari `opsworks.js` dengan menetapkan nilai dari db objek ke satu set `app.locals` properti, yang masing-masing memetakan ke salah satu elemen <%=... %> dalam file. `index.html` Dokumen yang dirender menggantikan elemen <%=... %> dengan nilai properti yang sesuai.

[3] Hubungkan ke instans Amazon RDS

Contoh ini digunakan `node-mysql` untuk mengakses database. Untuk terhubung ke database, contoh membuat `connection` objek dengan meneruskan data koneksi ke`createConnection`, dan kemudian memanggil `connection.connect` untuk membuat koneksi.

[4] Kueri database

Setelah membuat koneksi, contoh panggilan `connection.query` untuk query database. Contoh ini hanya query untuk nama database server. `query`mengembalikan array `results` objek, satu untuk setiap database, dengan nama database ditugaskan ke `Database` properti. Contoh menggabungkan nama dan menetapkannya yang menampilkan daftar di halaman `app.locals.databases,` HTML yang dirender.

Untuk contoh ini, ada lima database, `nodeexampledb` database yang Anda tentukan ketika Anda membuat instance RDS dan empat lainnya yang secara otomatis dibuat oleh Amazon RDS.

[5] Dengarkan permintaan yang masuk

Pernyataan akhir mendengarkan permintaan masuk pada port tertentu. Anda tidak perlu menentukan nilai port eksplisit. Saat menambahkan aplikasi ke tumpukan, Anda menentukan apakah aplikasi mendukung permintaan HTTP atau HTTPS. AWS OpsWorks Stacks kemudian menetapkan variabel `PORT` lingkungan ke 80 (HTTP) atau 443 (HTTPS), dan Anda dapat menggunakan variabel itu dalam aplikasi Anda.

Dimungkinkan untuk mendengarkan di port lain, tetapi grup keamanan bawaan lapisan Node.js App Server, AWS- OpsWorks -NodeJS-App-Server, memungkinkan lalu lintas pengguna masuk hanya ke port 80, 443, dan 22 (SSH). Untuk mengizinkan lalu lintas pengguna masuk ke port lain, buat grup keamanan dengan aturan masuk yang sesuai dan tetapkan ke lapisan Server Aplikasi Node.js. Jangan mengubah aturan masuk dengan mengedit grup keamanan bawaan. Setiap kali Anda membuat tumpukan, AWS OpsWorks Stacks menimpa grup keamanan bawaan dengan pengaturan standar, sehingga setiap perubahan yang Anda buat akan hilang.

> **Note**
>
> Anda dapat mengaitkan variabel lingkungan kustom dengan aplikasi Anda saat membuat atau memperbarui aplikasi terkait. Anda juga dapat meneruskan data ke aplikasi Anda dengan menggunakan JSON khusus dan resep khusus. Untuk informasi selengkapnya, lihat Melewati Data ke Aplikasi.

Membuat Server Database dan Load Balancer

Contoh ini menggunakan server database Amazon RDS dan instance penyeimbang beban Elastic Load Balancing. Anda harus membuat setiap instance secara terpisah dan kemudian memasukkannya ke dalam tumpukan Anda. Bagian ini menjelaskan cara membuat database baru dan instance penyeimbang beban. Sebagai gantinya, Anda dapat menggunakan instance yang ada, tetapi kami menyarankan Anda membaca prosedur untuk memastikan bahwa instans tersebut dikonfigurasi dengan benar.

Berikut ini menjelaskan cara membuat instance RDS DB yang dikonfigurasi minimal yang cukup untuk contoh ini. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon RDS.

Untuk membuat instance RDS DB

1. Buka konsol .

   Buka konsol Amazon RDS, dan atur wilayah ke US West (Oregon). Di panel navigasi, pilih Dasbor RDS, lalu pilih Luncurkan Instans DB.

2. Tentukan mesin database.

   Pilih MySQL Community Edition sebagai mesin database.

3. Tolak penerapan Multi-AZ.

   Pilih Tidak, contoh ini... , dan kemudian pilih Berikutnya. Anda tidak memerlukan penerapan Multi-AZ untuk contoh ini.

4. Konfigurasikan pengaturan dasar.

   Pada halaman Detail Instans DB, tentukan pengaturan berikut:

   - Kelas Instans DB: db.t2.micro
   - Penerapan Multi-AZ: Tidak
   - Penyimpanan yang Dialokasikan: GB **5**
   - Pengidentifikasi Instans DB: **nodeexample**
   - Nama Pengguna Master: **opsworksuser**
   - Master Password: Kata sandi pilihan Anda

   Rekam pengenal instans, nama pengguna, dan kata sandi untuk digunakan nanti, terima pengaturan default untuk opsi lain, lalu pilih Berikutnya.

5. Konfigurasikan pengaturan lanjutan.

   Pada halaman Konfigurasi Pengaturan Lanjut, tentukan pengaturan berikut:

   - Nama Database: **nodeexampledb**
   - Grup Keamanan DB: AWS- OpsWorks -DB-Master-Server

> **Note**
>
> Grup keamanan AWS- OpsWorks -DB-Master-Server hanya mengizinkan instans tumpukan Anda untuk mengakses database. Jika Anda ingin mengakses database secara langsung, lampirkan grup keamanan tambahan ke instans RDS DB dengan aturan masuk yang sesuai. Untuk informasi selengkapnya, lihat Grup Keamanan Amazon RDS. Anda juga dapat mengontrol akses dengan meletakkan instance di VPC. Untuk informasi selengkapnya, lihat Menjalankan Stack di VPC.

Rekam nama database untuk digunakan nanti, terima nilai default untuk pengaturan lain, lalu pilih Luncurkan Instans DB.

Prosedur berikut menjelaskan cara membuat penyeimbang beban Elastic Load Balancing untuk contoh ini. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna Penyeimbang Beban Elastis.

Untuk membuat penyeimbang beban

1. Buka EC2 konsol Amazon.

   Buka EC2 konsol Amazon dan pastikan wilayah tersebut diatur ke US West (Oregon). Di panel navigasi, pilih Load Balancers, lalu pilih Create Load Balancer.

2. Tentukan penyeimbang beban.

   Pada halaman Define Load Balancer, tentukan pengaturan berikut.

   - Nama – **`Node-LB`**
   - Buat LB Di Dalam - VPC Default Saya

   Terima pengaturan default untuk opsi lain, lalu pilih Berikutnya.

3. Tetapkan grup keamanan.

   Pada halaman Tetapkan Grup Keamanan, tentukan grup berikut:

   - grup keamanan VPC default

- AWS- OpsWorks -NodeJS-App-Server

Pilih Berikutnya. Pada halaman Konfigurasi Pengaturan Keamanan, pilih Berikutnya. Anda tidak memerlukan pendengar yang aman untuk contoh ini.

4. Konfigurasikan pemeriksaan kesehatan.

   Pada halaman Configure Health Check, atur Ping Path ke **/** dan terima nilai default untuk pengaturan lainnya. Pilih Berikutnya. Pada halaman Tambah EC2 Instans, pilih Berikutnya. Pada halaman Tambahkan Tag, pilih Tinjau dan Buat. AWS OpsWorks Stacks menangani tugas menambahkan EC2 instance ke penyeimbang beban, dan Anda tidak memerlukan tag untuk contoh ini.

5. Buat penyeimbang beban.

   Pada halaman Review, pilih Create to create the load balancer.

Membuat Stack

Anda sekarang memiliki semua komponen yang diperlukan untuk membuat tumpukan.

Untuk membuat tumpukan

1. Masuk ke konsol AWS OpsWorks Stacks.

   Masuk ke konsol AWS OpsWorks Stacks, dan pilih Add Stack.

2. Buat tumpukan.

   Untuk membuat tumpukan baru, pilih tumpukan Chef 11, lalu tentukan pengaturan berikut.

   - – **`NodeStack`**
   - Wilayah - AS Barat (Oregon)

     Anda dapat membuat tumpukan di wilayah AWS mana pun, tetapi kami merekomendasikan US West (Oregon) untuk tutorial.

   Pilih Tambahkan Tumpukan. Untuk informasi selengkapnya tentang pengaturan konfigurasi tumpukan, lihatMembuat Tumpukan Baru.

3. Tambahkan layer Server Aplikasi Node.js dengan penyeimbang beban terlampir.

   Pada NodeStackhalaman, pilih Tambahkan lapisan, lalu tentukan pengaturan berikut:

   - Jenis lapisan - Server Aplikasi Node.js
   - Elastic Load Balancer — Node-LB

   Terima nilai default untuk pengaturan lainnya, lalu pilih Tambah Lapisan.

4. Tambahkan instance ke layer dan mulai.

   Di panel navigasi, pilih Instans, lalu tambahkan dua instance ke layer Rails App Server, sebagai berikut.

   1. Di bawah Node.js App Server, pilih Tambahkan instance.

      Atur Ukuran ke t2.micro, terima nilai default untuk pengaturan lainnya, lalu pilih Tambahkan Instance.

   2. Pilih +Instance, dan kemudian tambahkan instance t2.micro kedua ke layer di subnet yang berbeda.

      Ini menempatkan instance di Availability Zone (AZ) yang berbeda.

   3. Pilih Tambahkan contoh.
   4. Untuk memulai kedua instance, pilih Mulai Semua Instans.

   Anda telah menetapkan penyeimbang beban Elastic Load Balancing ke layer ini. Ketika sebuah instance memasuki atau meninggalkan status online, AWS OpsWorks Stacks secara otomatis mendaftarkan atau membatalkan pendaftaran instance dengan penyeimbang beban.

   > **Note**
   >
   > Untuk tumpukan produksi, sebaiknya Anda mendistribusikan instance server aplikasi Anda di beberapa AZs. Jika pengguna tidak dapat terhubung ke AZ, penyeimbang beban merutekan lalu lintas masuk ke instans di zona yang tersisa, dan situs Anda akan terus berfungsi.

5. Daftarkan instans RDS DB dengan tumpukan.

   Di panel navigasi, pilih Resources dan daftarkan instans RDS DB dengan tumpukan, sebagai berikut.

   1. Pilih tab RDS, lalu pilih Tampilkan instans DB RDS Tidak Terdaftar.
   2. Pilih instance nodeexampledb, lalu tentukan pengaturan berikut:
      - User - Nama pengguna master yang Anda tentukan saat Anda membuat instance; untuk contoh ini. **`opsworksuser`**.
      - Kata sandi — Kata sandi utama yang Anda tentukan saat Anda membuat instance.
   3. Pilih Daftar dengan Stack untuk menambahkan instans RDS DB ke tumpukan sebagai lapisan layanan Amazon RDS.

   > **Warning**
   >
   > AWS OpsWorks Tumpukan tidak memvalidasi nilai Pengguna atau Kata Sandi, itu hanya meneruskannya ke aplikasi. Jika Anda salah memasukkannya, aplikasi Anda tidak dapat terhubung ke database.

   Untuk menambahkan instans RDS DB ke tumpukan sebagai lapisan layanan Amazon RDS, pilih Register with Stack.

Menyebarkan Aplikasi

Anda harus menyimpan aplikasi di repositori jarak jauh. Saat Anda menerapkannya, AWS OpsWorks Stacks menyebarkan kode dan file terkait dari repositori ke instance server aplikasi. Untuk kenyamanan, contoh ini menggunakan arsip Publik Amazon Simple Storage Service (Amazon S3) sebagai repositori, tetapi Anda juga dapat menggunakan beberapa jenis repositori lainnya, termasuk Git dan Subversion. Untuk informasi selengkapnya, lihat Sumber Aplikasi.

Untuk menyebarkan aplikasi

1. Package aplikasi dalam file arsip.

   Buat `.zip` arsip `nodedb` direktori dan subdirektori bernama nodedb.zip. Anda juga dapat menggunakan jenis file arsip lainnya, termasuk gzip, bzip2, dan tarball. Perhatikan bahwa

AWS OpsWorks Stacks tidak mendukung tarball yang tidak terkompresi. Untuk informasi selengkapnya, lihat Sumber Aplikasi.

2. Unggah file arsip ke Amazon S3.

   Unggah `nodedb.zip` ke bucket Amazon S3, buat file publik, dan salin URL file untuk digunakan nanti. Untuk informasi selengkapnya tentang cara membuat bucket dan mengunggah file, buka Memulai Dengan Amazon Simple Storage Service.

   > **Note**
   >
   > AWS OpsWorks Tumpukan juga dapat menyebarkan file pribadi dari bucket Amazon S3, tetapi untuk kesederhanaan, contoh ini menggunakan file publik. Untuk informasi selengkapnya, lihat Sumber Aplikasi.

3. Buat aplikasi AWS OpsWorks Stacks.

   Kembali ke konsol AWS OpsWorks Stacks, di panel navigasi, pilih Aplikasi, lalu pilih Tambahkan aplikasi. Tentukan pengaturan berikut:

   - Nama – `NodeDB`.

     String ini adalah nama tampilan aplikasi. Untuk sebagian besar tujuan, Anda memerlukan nama pendek aplikasi, yang dihasilkan AWS OpsWorks Stacks dari nama tampilan dengan mengubah semua karakter menjadi huruf kecil dan menghapus tanda baca. Untuk contoh ini, nama pendeknya adalah`nodedb`. Untuk memverifikasi nama pendek aplikasi, setelah membuat aplikasi, pilih aplikasi di halaman Aplikasi untuk menampilkan halaman detailnya.
   - Jenis —`Node.js`.
   - Jenis sumber data —`RDS`.
   - Instans database — Pilih instans Amazon RDS DB yang Anda daftarkan sebelumnya.
   - Nama database - Tentukan nama database yang Anda buat sebelumnya, `nodeexampledb` untuk contoh ini.
   - Jenis repositori —. `Http Archive`

     Anda harus menggunakan jenis repositori ini untuk file Amazon S3 publik. `S3 Archive`Jenis ini hanya digunakan untuk arsip pribadi.
   - URL Repositori - URL Amazon S3 file arsip.

Gunakan nilai default untuk setelan yang tersisa, lalu klik Tambah Aplikasi untuk membuat aplikasi.

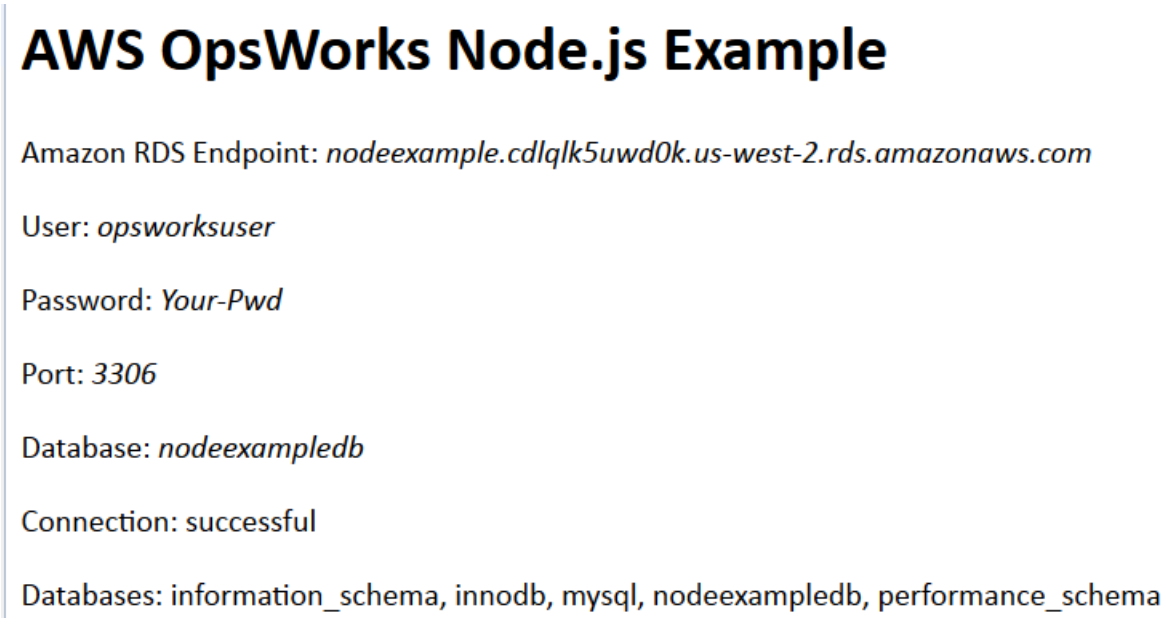
4. Terapkan aplikasi.

   Buka halaman Apps, dan di kolom Actions aplikasi NodeDB, pilih deploy. Kemudian pilih Deploy untuk menyebarkan aplikasi ke instance server. AWS OpsWorks Stacks menjalankan resep Deploy pada setiap instance, yang mengunduh aplikasi dari repositori dan memulai ulang server. Ketika setiap instance memiliki tanda centang hijau dan Status berhasil, penerapan selesai dan aplikasi siap untuk mulai menangani permintaan.

   > **Note**
   >
   > Jika penerapan gagal, pilih tampilkan di kolom Log untuk menampilkan log Chef penerapan. Informasi kesalahan berada di dekat bagian bawah.

5. Buka aplikasi .

   Untuk membuka aplikasi, pilih Layers, pilih load balancer, dan kemudian pilih nama DNS load balancer, yang mengirimkan permintaan HTTP ke load balancer. Anda harus melihat sesuatu seperti berikut ini.



> **Note**
>
> AWS OpsWorks Stacks secara otomatis menyebarkan aplikasi ke instance baru selama penyiapan. Penyebaran manual hanya diperlukan untuk instance online. Untuk informasi selengkapnya, lihat Menerapkan Aplikasi. Untuk diskusi umum tentang penyebaran, termasuk beberapa strategi penyebaran yang lebih canggih, lihat. Mengelola dan Menyebarkan Aplikasi dan Buku Masak

Apa Selanjutnya?

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Panduan ini membawa Anda melalui dasar-dasar pengaturan tumpukan server aplikasi Node.js sederhana. Berikut adalah beberapa saran untuk apa yang harus dilakukan selanjutnya.

Periksa buku masak bawaan Node.js

Jika Anda ingin mengetahui bagaimana instance dikonfigurasi secara detail, lihat buku masak bawaan lapisan, opsworks_nodejs, yang berisi resep dan file terkait yang digunakan AWS OpsWorks Stacks untuk menginstal dan mengonfigurasi perangkat lunak, dan buku masak penerapan bawaan, yang berisi resep yang digunakan Stacks untuk menyebarkan aplikasi. AWS OpsWorks

Sesuaikan konfigurasi server

Contoh tumpukan cukup mendasar. Untuk penggunaan produksi, Anda mungkin ingin menyesuaikan tumpukan. Untuk informasi selengkapnya, lihat Menyesuaikan Tumpukan AWS OpsWorks.

Tambahkan dukungan SSL

Anda dapat mengaktifkan dukungan SSL untuk aplikasi Anda dan memberikan AWS OpsWorks Stacks dengan sertifikat yang sesuai saat Anda membuat aplikasi. AWS OpsWorks Tumpukan

kemudian menginstal sertifikat di direktori yang sesuai. Untuk informasi selengkapnya, lihat Menggunakan SSL.

Tambahkan cache data dalam memori

Situs tingkat produksi sering meningkatkan kinerja dengan menyimpan data dalam penyimpanan nilai kunci dalam memori, seperti Redis atau Memcache. Anda dapat menggunakan keduanya dengan tumpukan AWS OpsWorks Stacks. Untuk informasi selengkapnya, lihat ElastiCache Redis dan Memcached.

Gunakan strategi penyebaran yang lebih canggih

Contoh ini menggunakan strategi penerapan aplikasi sederhana, yang menerapkan pembaruan ke setiap instance secara bersamaan. Pendekatan ini sederhana dan cepat, tetapi tidak ada margin untuk kesalahan. Jika penerapan gagal atau pembaruan memiliki masalah, setiap instance di tumpukan produksi Anda dapat terpengaruh, berpotensi mengganggu atau menonaktifkan situs Anda hingga Anda dapat memperbaiki masalah. Untuk informasi selengkapnya tentang strategi penerapan, lihatMengelola dan Menyebarkan Aplikasi dan Buku Masak.

Memperluas lapisan Server Aplikasi Node.js

Anda dapat memperluas lapisan dengan berbagai cara. Misalnya, Anda dapat menerapkan resep untuk menjalankan skrip pada instance atau mengimplementasikan kait penerapan Chef untuk menyesuaikan penerapan aplikasi. Untuk informasi selengkapnya, lihat Memperluas Layer.

Tentukan variabel lingkungan

Anda dapat meneruskan data ke aplikasi Anda dengan menentukan variabel lingkungan untuk aplikasi terkait. Saat menerapkan aplikasi, AWS OpsWorks Stacks akan mengekspor variabel tersebut sehingga Anda dapat mengaksesnya dari aplikasi. Untuk informasi selengkapnya, lihat Menggunakan Variabel Lingkungan.

Menyesuaikan Tumpukan AWS OpsWorks

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks built-in layers menyediakan fungsionalitas standar yang cukup untuk berbagai tujuan. Namun, Anda mungkin menemukan satu atau lebih hal berikut:

- Konfigurasi standar lapisan bawaan memadai tetapi tidak ideal; Anda ingin mengoptimalkannya untuk kebutuhan khusus Anda.

  Misalnya, Anda mungkin ingin menyetel konfigurasi server Nginx pada lapisan Server Web Statis dengan menentukan nilai Anda sendiri untuk pengaturan seperti jumlah maksimum proses pekerja atau nilainya. `keepalivetimeout`
- Fungsionalitas lapisan bawaan baik-baik saja, tetapi Anda ingin memperluasnya dengan menginstal paket tambahan atau menjalankan beberapa skrip instalasi khusus.

  Misalnya, Anda mungkin ingin memperluas lapisan PHP App Server dengan juga menginstal server Redis.
- Anda memiliki persyaratan yang tidak ditangani oleh salah satu lapisan bawaan.

  Misalnya, AWS OpsWorks Stacks tidak menyertakan lapisan bawaan untuk beberapa server basis data populer. Anda dapat membuat lapisan kustom yang menginstal server tersebut pada instance layer.
- Anda menjalankan tumpukan Windows, yang hanya mendukung lapisan khusus.

AWS OpsWorks Stacks menyediakan berbagai cara untuk menyesuaikan lapisan untuk memenuhi kebutuhan spesifik Anda. Contoh-contoh berikut tercantum dalam urutan peningkatan kompleksitas dan kekuatan:

> **Note**
>
> Beberapa pendekatan ini hanya berfungsi untuk tumpukan Linux. Lihat topik berikut untuk detailnya.

- Gunakan JSON khusus untuk mengganti pengaturan AWS OpsWorks Stacks default.
- Terapkan buku masak Chef kustom dengan file atribut yang mengganti pengaturan Stacks default AWS OpsWorks .
- Menerapkan buku masak Chef kustom dengan template yang menggantikan atau memperluas template Stacks default. AWS OpsWorks
- Terapkan buku masak Chef khusus dengan resep sederhana yang menjalankan skrip shell.

- Menerapkan buku masak Chef kustom dengan resep yang melakukan tugas-tugas seperti membuat dan mengonfigurasi direktori, menginstal paket, membuat file konfigurasi, menyebarkan aplikasi, dan sebagainya.

Anda juga dapat mengganti resep, tergantung pada versi dan sistem operasi Chef stack.

- Dengan tumpukan Chef 0.9 dan 11.4, Anda tidak dapat mengganti resep bawaan dengan menerapkan resep khusus dengan buku masak dan nama resep yang sama.

  Untuk setiap acara siklus hidup, AWS OpsWorks Stacks selalu menjalankan resep bawaan terlebih dahulu, diikuti oleh resep khusus apa pun. Karena versi Chef ini tidak menjalankan resep dengan buku masak dan nama resep yang sama dua kali, resep bawaan diutamakan dan resep khusus tidak dijalankan.
- Anda dapat mengganti resep bawaan di tumpukan Chef 11.10.

  Untuk informasi selengkapnya, lihat Instalasi dan Prioritas Buku Masak.
- Anda tidak dapat mengganti resep bawaan di tumpukan Windows.

  Cara AWS OpsWorks Stacks menangani Chef berjalan untuk tumpukan Windows tidak memungkinkan resep bawaan diganti.

> **Note**
>
> Karena banyak teknik menggunakan buku masak khusus, Anda harus membaca terlebih dahulu Buku Masak dan Resep jika Anda belum terbiasa dengan implementasi buku masak. Dasar-dasar Buku Masakmemberikan pengantar tutorial terperinci untuk mengimplementasikan buku masak khusus, dan Menerapkan Buku Masak untuk Tumpukan AWS OpsWorks mencakup beberapa detail tentang cara menerapkan buku masak untuk instance AWS OpsWorks Stacks.

Topik

- Menyesuaikan Konfigurasi AWS OpsWorks Tumpukan dengan Mengganti Atribut
- Memperluas File Konfigurasi AWS OpsWorks Tumpukan Menggunakan Template Kustom
- Memperluas Layer
- Membuat Layer Server Tomcat Kustom

- Konfigurasi Stack dan Atribut Deployment

Menyesuaikan Konfigurasi AWS OpsWorks Tumpukan dengan Mengganti Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Untuk tumpukan Windows dan tumpukan Linux Chef 12, AWS OpsWorks Stacks menggunakan Chef yang dijalankan terpisah untuk resep bawaan dan resep khusus. Ini berarti Anda tidak dapat menggunakan teknik yang dibahas di bagian ini untuk mengganti atribut bawaan untuk tumpukan Windows dan tumpukan Linux Chef 12.

Resep dan template bergantung pada berbagai atribut Chef misalnya atau informasi spesifik tumpukan seperti konfigurasi lapisan atau pengaturan server aplikasi. Atribut ini memiliki beberapa sumber:

- JSON Kustom —Anda dapat secara opsional menentukan atribut JSON kustom saat membuat, memperbarui, atau mengkloning tumpukan, atau saat Anda menerapkan aplikasi.
- Atribut konfigurasi tumpukan —AWS OpsWorks Stacks mendefinisikan atribut ini untuk menyimpan informasi konfigurasi tumpukan, termasuk informasi yang Anda tentukan melalui pengaturan konsol.
- Atribut penerapan —AWS OpsWorks mendefinisikan atribut terkait penerapan untuk peristiwa Deploy.
- Atribut buku masak — Cookbook bawaan dan kustom biasanya menyertakan satu atau lebih file atribut, yang berisi atribut yang mewakili nilai khusus buku masak seperti pengaturan konfigurasi server aplikasi.

- Chef —Chef's Ohai tool mendefinisikan atribut yang mewakili berbagai pengaturan konfigurasi sistem, seperti tipe CPU dan memori yang diinstal.

Untuk daftar lengkap konfigurasi tumpukan dan atribut penerapan dan atribut buku masak bawaan, lihat Konfigurasi Stack dan Atribut Deployment: Linux dan. Atribut Buku Masak Bawaan Untuk informasi lebih lanjut tentang atribut Ohai, lihat Ohai.

Ketika peristiwa siklus hidup seperti Deploy atau Configure terjadi, atau Anda menjalankan perintah stack seperti `execute_recipes` atau`update_packages`, AWS OpsWorks Stacks melakukan hal berikut:

- Mengirim perintah yang sesuai ke agen pada setiap instance yang terpengaruh.

  Agen menjalankan resep yang sesuai. Misalnya, untuk acara Deploy, agen menjalankan resep Deploy bawaan, diikuti oleh resep Deploy khusus apa pun.
- Menggabungkan atribut JSON dan penerapan kustom dengan atribut konfigurasi tumpukan dan menginstalnya pada instance.

Atribut dari JSON kustom, konfigurasi tumpukan dan atribut penerapan, atribut buku masak, dan atribut Ohai digabungkan menjadi objek simpul, yang memasok nilai atribut ke resep. Sebuah instance pada dasarnya stateless sejauh menyangkut atribut konfigurasi tumpukan, termasuk JSON khusus apa pun. Saat Anda menjalankan perintah deployment atau stack, resep terkait menggunakan atribut konfigurasi stack yang diunduh dengan perintah tersebut.

Topik

- Atribut Prioritas
- Mengganti Atribut Dengan JSON Kustom
- Mengganti Atribut AWS OpsWorks Stacks Menggunakan Atribut Cookbook Kustom

Atribut Prioritas

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

> Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika atribut didefinisikan secara unik, Chef hanya memasukkannya ke dalam objek node. Namun, sumber atribut apa pun dapat menentukan atribut apa pun, sehingga dimungkinkan untuk atribut yang sama memiliki beberapa definisi dengan nilai yang berbeda. Misalnya, `apache2` buku masak bawaan mendefinisikan`node[:apache][:keepalive]`, tetapi Anda juga dapat menentukan atribut itu di JSON khusus atau di buku masak khusus. Jika atribut memiliki beberapa definisi, mereka dievaluasi dalam urutan yang dijelaskan kemudian dan objek node menerima definisi dengan prioritas tertinggi.

Atribut didefinisikan sebagai berikut:

```
node.type[:attribute][:sub_attribute][:...]=value
```

Jika atribut memiliki beberapa definisi, tipe menentukan definisi mana yang diutamakan, dan definisi tersebut dimasukkan ke dalam objek node. AWS OpsWorks Tumpukan menggunakan jenis atribut berikut:

- default —Ini adalah tipe yang paling umum, dan pada dasarnya berarti “gunakan nilai ini jika atribut belum ditentukan.” Jika semua definisi atribut adalah `default` tipe, definisi pertama dalam urutan evaluasi diutamakan dan nilai selanjutnya diabaikan. Perhatikan bahwa AWS OpsWorks Stacks menetapkan semua konfigurasi tumpukan dan definisi atribut penerapan untuk `default` mengetik.
- normal —Atribut dengan tipe ini mengesampingkan `normal` atribut apa pun `default` atau yang didefinisikan sebelumnya dalam urutan evaluasi. Misalnya, jika atribut pertama berasal dari buku masak bawaan dan memiliki `default` tipe dan yang kedua adalah atribut yang ditentukan pengguna dengan memiliki `normal` tipe, definisi kedua diutamakan.
- set —Ini adalah tipe usang yang mungkin Anda lihat di buku masak lama. Itu telah digantikan oleh`normal`, yang memiliki prioritas yang sama.

Chef mendukung beberapa jenis atribut tambahan, termasuk `automatic` tipe yang lebih diutamakan daripada semua definisi atribut lainnya. Definisi atribut yang dihasilkan oleh alat Ohai Chef adalah semua `automatic` jenis, sehingga mereka secara efektif hanya-baca. Ini biasanya tidak menjadi masalah, karena tidak ada alasan untuk mengesampingkannya dan mereka berbeda dari atribut AWS OpsWorks Stacks. Namun, Anda harus berhati-hati untuk memberi nama atribut buku masak

khusus Anda sehingga mereka berbeda dari atribut Ohai. Untuk informasi selengkapnya, lihat Tentang Atribut.

> **Note**
>
> Alat Ohai adalah executable yang dapat Anda jalankan dari baris perintah. Untuk mencantumkan atribut Ohai instance, masuk ke instance dan jalankan `ohai` di jendela terminal. Ketahuilah bahwa itu menghasilkan output yang sangat panjang.

Berikut adalah langkah-langkah yang menggabungkan berbagai definisi atribut ke dalam objek node:

1. Gabungkan atribut konfigurasi tumpukan khusus apa pun ke dalam konfigurasi tumpukan dan atribut penerapan.

   Atribut JSON kustom dapat diatur untuk tumpukan, atau untuk penerapan tertentu. Mereka adalah yang pertama dalam urutan evaluasi dan merupakan `normal` tipe yang efektif. Jika satu atau beberapa atribut konfigurasi tumpukan juga didefinisikan dalam JSON kustom, nilai JSON kustom diutamakan. Jika tidak, AWS OpsWorks Stacks hanya memasukkan atribut JSON khusus ke dalam konfigurasi tumpukan.
2. Gabungkan atribut JSON kustom penerapan apa pun ke dalam konfigurasi tumpukan dan atribut penerapan.

   Atribut JSON kustom penerapan juga merupakan `normal` tipe yang efektif, sehingga lebih diutamakan daripada konfigurasi tumpukan bawaan dan kustom JSON dan JSON penerapan bawaan.
3. Gabungkan konfigurasi tumpukan dan atribut penerapan ke dalam objek node instance.
4. Gabungkan atribut buku masak bawaan instance ke dalam objek node.

   Atribut buku masak bawaan adalah semua `default` jenis. Jika satu atau beberapa atribut buku masak bawaan juga ditentukan dalam konfigurasi tumpukan dan atribut penerapan — biasanya karena Anda mendefinisikannya dengan JSON kustom — definisi konfigurasi tumpukan lebih diutamakan daripada definisi buku masak bawaan. Semua atribut buku masak bawaan lainnya hanya dimasukkan ke dalam objek node.
5. Gabungkan atribut buku masak khusus instance ke dalam objek node.

   Atribut kustom c ookbook biasanya salah satu `normal` atau `default` tipe. Atribut unik dimasukkan ke dalam objek node. Jika atribut buku masak kustom juga didefinisikan dalam

Langkah 1-3 (biasanya karena Anda mendefinisikannya dengan JSON kustom), prioritas tergantung pada jenis atribut buku masak kustom:

- Atribut yang didefinisikan dalam Langkah 1-3 lebih diutamakan daripada atribut buku masak khusus. `default`
- `normal`Atribut ookbook c kustom lebih diutamakan daripada definisi dari Langkah 1-3.

> **Important**
>
> Jangan gunakan `default` atribut buku masak khusus untuk mengganti konfigurasi tumpukan atau atribut buku masak bawaan. Karena atribut buku masak kustom dievaluasi terakhir, `default` atribut memiliki prioritas terendah, dan tidak dapat mengganti apa pun.

Mengganti Atribut Dengan JSON Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Karena AWS OpsWorks Stacks menangani Chef berjalan berbeda untuk tumpukan Windows daripada untuk tumpukan Linux, Anda tidak dapat menggunakan teknik yang dibahas di bagian ini untuk tumpukan Windows.

Cara termudah untuk mengganti atribut AWS OpsWorks Stacks adalah dengan mendefinisikannya di JSON khusus, yang lebih diutamakan daripada konfigurasi tumpukan dan atribut penerapan serta atribut buku masak bawaan dan khusus. `default` Untuk informasi selengkapnya, lihat Atribut Prioritas.

> **Important**
>
> Anda harus mengganti konfigurasi tumpukan dan atribut penerapan dengan hati-hati. Misalnya, mengganti atribut di `opsworks` namespace dapat mengganggu resep bawaan. Untuk informasi selengkapnya, lihat Konfigurasi Stack dan Atribut Deployment.

Anda juga dapat menggunakan JSON khusus untuk menentukan atribut unik, biasanya untuk meneruskan data ke resep kustom Anda. Atribut hanya dimasukkan ke dalam objek node, dan resep dapat mereferensikannya dengan menggunakan sintaks simpul Chef standar.

Cara Menentukan JSON Kustom

Untuk menggunakan JSON kustom untuk mengganti nilai atribut, Anda harus terlebih dahulu menentukan nama atribut atribut yang sepenuhnya memenuhi syarat. Anda kemudian membuat objek JSON yang berisi atribut yang ingin Anda timpa, disetel ke nilai pilihan Anda. Untuk kenyamanan, Konfigurasi Stack dan Atribut Deployment: Linux dan Atribut Buku Masak Bawaan dokumen yang biasa digunakan konfigurasi tumpukan, penerapan, dan atribut buku masak bawaan, termasuk nama yang sepenuhnya memenuhi syarat.

Hubungan orang tua-anak objek harus sesuai dengan node Chef yang memenuhi syarat penuh yang sesuai. Misalnya, Anda ingin mengubah atribut Apache berikut:

- `keepalivetimeout`Atribut, yang node adalah `node[:apache][:keepalivetimeout]` dan memiliki nilai default`3`.
- `logrotate``schedule`Atribut, yang simpulnya`node[:apache][:logrotate][:schedule]`, dan memiliki nilai default`"daily"`.

Untuk mengganti atribut dan menetapkan nilai ke `5` dan`"weekly"`, masing-masing, Anda akan menggunakan JSON kustom berikut:

```
{
  "apache" : {
    "keepalivetimeout" : 5,
    "logrotate" : {
       "schedule" : "weekly"
    }
  }
```

```
}
```

Kapan Menentukan JSON Kustom

Anda dapat menentukan struktur JSON kustom untuk tugas-tugas berikut:

- Buat tumpukan baru
- Perbarui tumpukan
- Jalankan perintah tumpukan
- Kloning tumpukan
- Menerapkan aplikasi

Untuk setiap tugas, AWS OpsWorks Stacks menggabungkan atribut JSON kustom dengan konfigurasi tumpukan dan atribut penerapan dan mengirimkannya ke instance, untuk digabungkan ke objek node. Namun, perhatikan hal berikut:

- Jika Anda menentukan JSON kustom saat membuat, mengkloning, atau memperbarui tumpukan, atribut digabungkan ke dalam konfigurasi tumpukan dan atribut penerapan untuk semua peristiwa siklus hidup berikutnya dan perintah tumpukan.
- Jika Anda menentukan JSON kustom untuk penerapan, atribut digabungkan ke dalam konfigurasi tumpukan dan atribut penerapan hanya untuk peristiwa yang sesuai.

  Jika Anda ingin menggunakan atribut kustom tersebut untuk penerapan berikutnya, Anda harus secara eksplisit menentukan JSON kustom lagi.

Penting untuk diingat bahwa atribut hanya memengaruhi instance ketika digunakan oleh resep. Jika Anda mengganti nilai atribut tetapi tidak ada resep berikutnya yang mereferensikan atribut, perubahan tidak berpengaruh. Anda harus memastikan bahwa JSON kustom dikirim sebelum resep terkait dijalankan, atau memastikan bahwa resep yang sesuai dijalankan kembali.

Praktik Terbaik JSON Kustom

Anda dapat menggunakan JSON khusus untuk mengganti atribut AWS OpsWorks Stacks apa pun, tetapi memasukkan informasi secara manual agak rumit, dan tidak berada di bawah kontrol sumber apa pun. Kustom JSON paling baik digunakan untuk tujuan berikut:

- Ketika Anda ingin mengganti hanya sejumlah kecil atribut, dan Anda tidak perlu menggunakan buku masak khusus.

Dengan JSON khusus, Anda dapat menghindari overhead pengaturan dan pemeliharaan repositori buku masak hanya untuk mengganti beberapa atribut.

- Nilai sensitif, seperti kata sandi atau kunci otentikasi.

  Atribut buku masak disimpan dalam repositori, sehingga informasi sensitif apa pun berisiko dikompromikan. Sebagai gantinya, tentukan atribut dengan nilai dummy dan gunakan JSON khusus untuk mengatur nilai sebenarnya.

- Nilai yang diharapkan bervariasi.

  Misalnya, praktik yang disarankan adalah agar tumpukan produksi Anda didukung oleh tumpukan pengembangan dan pementasan yang terpisah. Misalkan tumpukan ini mendukung aplikasi yang menerima pembayaran. Jika Anda menggunakan JSON khusus untuk menentukan titik akhir pembayaran, Anda dapat menentukan URL pengujian untuk tumpukan pementasan Anda. Saat Anda siap untuk memigrasikan tumpukan yang diperbarui ke tumpukan produksi, Anda dapat menggunakan buku masak yang sama dan menggunakan JSON khusus untuk mengatur titik akhir pembayaran ke URL produksi.

- Nilai yang spesifik untuk perintah stack atau deployment tertentu.

Mengganti Atribut AWS OpsWorks Stacks Menggunakan Atribut Cookbook Kustom

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Untuk tumpukan Windows, AWS OpsWorks Stacks menggunakan Chef yang dijalankan terpisah untuk resep bawaan dan resep khusus. Ini berarti Anda tidak dapat menggunakan teknik yang dibahas di bagian ini untuk mengganti atribut bawaan untuk tumpukan Windows.

Custom JSON adalah cara mudah untuk mengganti konfigurasi tumpukan AWS OpsWorks Stacks dan atribut buku masak bawaan, tetapi memiliki beberapa keterbatasan. Secara khusus, Anda harus memasukkan JSON khusus secara manual untuk setiap penggunaan, sehingga Anda tidak memiliki cara yang kuat untuk mengelola definisi. Pendekatan yang lebih baik adalah sering menggunakan file atribut buku masak khusus untuk mengganti atribut bawaan. Melakukannya memungkinkan Anda untuk menempatkan definisi di bawah kendali sumber.

Prosedur untuk menggunakan file atribut khusus untuk mengganti definisi AWS OpsWorks Stacks sangat mudah.

Untuk mengganti definisi atribut AWS OpsWorks Stacks

1. Siapkan repositori buku masak, seperti yang dijelaskan dalam. Buku Masak dan Resep
2. Buat buku masak dengan nama yang sama dengan buku masak bawaan yang berisi atribut yang ingin Anda timpa. Misalnya, untuk mengganti atribut Apache, buku masak harus diberi nama apache2.
3. Tambahkan `attributes` folder ke buku masak dan tambahkan file ke folder bernama `customize.rb` itu.
4. Tambahkan definisi atribut ke file untuk setiap atribut buku masak bawaan yang ingin Anda timpa, setel ke nilai pilihan Anda. Atribut harus `normal` tipe atau lebih tinggi dan memiliki nama node yang sama persis dengan atribut AWS OpsWorks Stacks yang sesuai. Untuk daftar detail atribut AWS OpsWorks Stacks, termasuk nama node, lihat Konfigurasi Stack dan Atribut Deployment: Linux danAtribut Buku Masak Bawaan. Untuk informasi selengkapnya tentang atribut dan atribut file, lihat Tentang File Atribut.

   > **Important**
   >
   > Atribut Anda harus `normal` tipe untuk mengganti atribut AWS OpsWorks Stacks; `default` tipe tidak diutamakan. Misalnya, jika `customize.rb` file Anda berisi definisi `default[:apache][:keepalivetimeout] = 5` atribut, atribut yang sesuai dalam file `apache.rb` atribut bawaan dievaluasi terlebih dahulu, dan diutamakan. Untuk informasi selengkapnya, lihat Mengesampingkan Atribut.

5. Ulangi Langkah 2 - 4 untuk setiap buku masak bawaan dengan atribut yang ingin Anda timpa.
6. Aktifkan buku masak khusus untuk tumpukan Anda dan berikan informasi yang diperlukan untuk AWS OpsWorks Stacks untuk mengunduh buku masak Anda ke instance tumpukan. Untuk informasi selengkapnya, lihat Memasang Buku Masak Kustom.

> **Note**
>
> Untuk panduan lengkap tentang prosedur ini, lihat. Mengesampingkan Atribut Bawaan

Objek node yang digunakan oleh peristiwa siklus hidup berikutnya, perintah penerapan, dan perintah tumpukan sekarang akan berisi definisi atribut Anda, bukan nilai Stacks. AWS OpsWorks

Misalnya, untuk mengganti Apache bawaan `keepalivetimeout` dan `logrotate schedule` pengaturan yang dibahas diCara Menentukan JSON Kustom, tambahkan `apache2` buku masak ke repositori Anda dan tambahkan `customize.rb` file ke folder buku masak dengan konten berikut. `attributes`

```
normal[:apache][:keepalivetimeout] = 5
normal[:apache][:logrotate][:schedule] = 'weekly'
```

> **Important**
>
> Anda tidak boleh mengganti atribut AWS OpsWorks Stacks dengan memodifikasi salinan file atribut bawaan terkait. Jika, misalnya, Anda menyalin `apache.rb` ke `apache2/attributes` folder Anda dan memodifikasi beberapa pengaturannya, Anda pada dasarnya mengganti setiap atribut dalam file bawaan. Resep akan menggunakan definisi atribut dari salinan Anda dan mengabaikan file bawaan. Jika AWS OpsWorks Stacks kemudian memodifikasi file atribut bawaan, resep tidak akan memiliki akses ke perubahan kecuali Anda memperbarui salinan Anda secara manual.
>
> Untuk menghindari situasi ini, semua buku masak bawaan berisi file `customize.rb` atribut kosong, yang diperlukan di semua modul melalui `include_attribute` arahan. Dengan mengganti atribut dalam salinan Anda`customize.rb`, Anda hanya memengaruhi atribut spesifik tersebut. Resep akan mendapatkan nilai atribut lain dari file atribut bawaan, dan secara otomatis mendapatkan nilai saat ini dari atribut apa pun yang belum Anda ganti. Pendekatan ini membantu Anda menjaga jumlah atribut di repositori buku masak Anda tetap kecil, yang mengurangi overhead pemeliharaan Anda dan membuat upgrade future lebih mudah dikelola.

Memperluas File Konfigurasi AWS OpsWorks Tumpukan Menggunakan Template Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Karena AWS OpsWorks Stacks menangani Chef berjalan berbeda untuk tumpukan Windows daripada untuk tumpukan Linux, Anda tidak dapat menggunakan teknik yang dibahas di bagian ini untuk tumpukan Windows.

AWS OpsWorks Stacks menggunakan template untuk membuat file seperti file konfigurasi, yang biasanya bergantung pada atribut untuk banyak pengaturan. Jika Anda menggunakan atribut JSON kustom atau buku masak khusus untuk mengganti definisi AWS OpsWorks Stacks, pengaturan pilihan Anda dimasukkan ke dalam file konfigurasi sebagai pengganti pengaturan Stacks. AWS OpsWorks Namun, AWS OpsWorks Stacks tidak selalu menentukan atribut untuk setiap pengaturan konfigurasi yang mungkin; ia menerima default untuk beberapa pengaturan dan hardcode lainnya langsung di template. Anda tidak dapat menggunakan atribut JSON kustom atau buku masak khusus untuk menentukan pengaturan yang disukai jika tidak ada atribut AWS OpsWorks Stacks yang sesuai.

Anda dapat memperluas file konfigurasi untuk menyertakan pengaturan konfigurasi tambahan dengan membuat template khusus. Anda kemudian dapat menambahkan pengaturan konfigurasi apa pun atau konten lain yang Anda butuhkan ke file, dan mengganti pengaturan hardcode apa pun. Untuk informasi lebih lanjut tentang template, lihatTemplate.

> **Note**
>
> Anda dapat mengganti template bawaan apa pun kecuali opsworks-agent.monitrc.erb.

Untuk membuat template kustom

1. Buat buku masak dengan struktur dan nama direktori yang sama dengan buku masak bawaan. Kemudian, buat file template di direktori yang sesuai dengan nama yang sama dengan template bawaan yang ingin Anda sesuaikan. Misalnya, untuk menggunakan template kustom untuk memperluas file `httpd.conf` konfigurasi Apache, Anda harus menerapkan `apache2` buku masak di repositori Anda dan file template Anda harus. `apache2/templates/default/apache.conf.erb` Menggunakan nama yang persis sama memungkinkan AWS OpsWorks Stacks mengenali template khusus dan menggunakannya alih-alih templat bawaan.

   Pendekatan paling sederhana adalah dengan hanya menyalin file template bawaan dari GitHubrepositori buku masak bawaan ke buku masak Anda dan memodifikasinya sesuai kebutuhan.

   > **Important**
   >
   > Jangan menyalin file apa pun dari buku masak bawaan kecuali untuk file templat yang ingin Anda sesuaikan. Salinan jenis file buku masak lainnya, seperti resep, membuat sumber daya Chef duplikat dan dapat menyebabkan kesalahan.

   Buku masak juga dapat menyertakan atribut khusus, resep, dan file terkait, tetapi nama file mereka tidak boleh menduplikasi nama file bawaan.
2. Sesuaikan file template untuk menghasilkan file konfigurasi yang memenuhi kebutuhan Anda. Anda dapat menambahkan lebih banyak pengaturan, menghapus pengaturan yang ada, mengganti atribut hardcode, dan sebagainya.
3. Jika Anda belum melakukannya, edit pengaturan tumpukan untuk mengaktifkan buku masak khusus dan tentukan repositori buku masak Anda. Untuk informasi selengkapnya, lihat Memasang Buku Masak Kustom.

> **Note**
>
> Untuk panduan lengkap tentang prosedur ini, lihat. Mengesampingkan Template Bawaan

Anda tidak perlu menerapkan resep apa pun atau menambahkan resep ke konfigurasi lapisan untuk mengganti template. AWS OpsWorks Tumpukan selalu menjalankan resep bawaan. Ketika

menjalankan resep yang membuat file konfigurasi, itu akan secara otomatis menggunakan template kustom Anda alih-alih template bawaan.

> **Note**
>
> Jika AWS OpsWorks Stacks membuat perubahan pada template bawaan, template kustom Anda mungkin tidak sinkron dan tidak lagi berfungsi dengan benar. Misalnya, anggaplah template Anda mengacu pada file dependen, dan nama file berubah. AWS OpsWorks Tumpukan tidak sering membuat perubahan seperti itu, dan ketika template berubah, itu mencantumkan perubahan dan memberi Anda opsi untuk meningkatkan ke versi baru. Anda harus memantau repositori AWS OpsWorks Stacks untuk perubahan, dan memperbarui template Anda secara manual sesuai kebutuhan.

Memperluas Layer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Terkadang, Anda perlu menyesuaikan lapisan bawaan di luar apa yang dapat ditangani dengan memodifikasi atribut AWS OpsWorks Stacks atau menyesuaikan templat. Misalnya, Anda perlu membuat symlink, mengatur mode file atau folder, menginstal paket tambahan, dan sebagainya. Anda harus memperluas lapisan khusus untuk menyediakan lebih dari fungsionalitas minimal. Dalam hal ini, Anda perlu mengimplementasikan satu atau lebih buku masak khusus dengan resep untuk menangani tugas penyesuaian. Topik ini memberikan beberapa contoh cara menggunakan resep untuk memperluas lapisan.

Jika Anda baru mengenal Chef, Anda harus membaca terlebih dahuluBuku masak 101, yang merupakan tutorial yang memperkenalkan dasar-dasar bagaimana menerapkan buku masak untuk melakukan berbagai tugas umum. Untuk contoh rinci tentang cara menerapkan lapisan kustom, lihatMembuat Layer Server Tomcat Kustom.

Topik

- Menggunakan Resep untuk Menjalankan Skrip
- Menggunakan Chef Deployment Hooks
- Menjalankan Cron Jobs di Instans Linux
- Menginstal dan Mengkonfigurasi Paket di Instans Linux

Menggunakan Resep untuk Menjalankan Skrip

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika Anda sudah memiliki skrip yang melakukan tugas kustomisasi yang diperlukan, pendekatan paling sederhana untuk memperluas lapisan sering kali menerapkan resep sederhana untuk menjalankan skrip. Anda kemudian dapat menetapkan resep ke peristiwa siklus hidup yang sesuai, biasanya Setup atau Deploy, atau menggunakan perintah `execute_recipes` stack untuk menjalankan resep secara manual.

Contoh berikut menjalankan skrip shell pada instance Linux, tetapi Anda dapat menggunakan pendekatan yang sama untuk jenis skrip lainnya, termasuk PowerShell skrip 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
```

`cookbook_file`Sumber daya mewakili file yang disimpan dalam subdirektori direktori buku masak, dan mentransfer file ke lokasi tertentu pada instance. `files` Contoh ini mentransfer skrip shell`lib-`

`installer.sh`,, ke `/tmp` direktori instance dan menetapkan mode file ke`0755`. Untuk informasi selengkapnya, lihat cookbook_file.

`execute`Sumber daya mewakili perintah, seperti perintah shell. Contoh ini berjalan`lib-installer.sh`. Untuk informasi selengkapnya, lihat mengeksekusi.

Anda juga dapat menjalankan skrip dengan memasukkannya ke dalam resep. Contoh berikut menjalankan skrip bash, tetapi Chef juga mendukung Csh, Perl, Python, dan Ruby.

```
script "install_something" do
  interpreter "bash"
  user "root"
  cwd "/tmp"
  code <<-EOH
    #insert bash script
  EOH
end
```

Sumber `script` daya mewakili skrip. Contoh menentukan interpreter bash, menetapkan pengguna ke`"root"`, dan menetapkan direktori kerja ke. `/tmp` Kemudian menjalankan skrip bash di `code` blok, yang dapat mencakup baris sebanyak yang diperlukan. Untuk informasi lebih lanjut, lihat skrip.

Untuk informasi selengkapnya tentang cara menggunakan resep untuk menjalankan skrip, lihatContoh 7: Menjalankan Perintah dan Skrip. Untuk contoh cara menjalankan PowerShell skrip pada instance Windows, lihatMenjalankan PowerShell Skrip Windows.

Menggunakan Chef Deployment Hooks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menyesuaikan penerapan dengan menerapkan resep khusus untuk melakukan tugas yang diperlukan dan menetapkannya ke acara Deploy layer yang sesuai. Pendekatan alternatif dan

terkadang lebih sederhana—terutama jika Anda tidak perlu mengimplementasikan buku masak untuk tujuan lain—adalah dengan menggunakan kait penerapan Chef untuk menjalankan kode kustomisasi Anda. Selain itu, resep Deploy khusus dijalankan setelah penerapan dilakukan oleh resep bawaan. Deployment hook memungkinkan Anda untuk berinteraksi selama penerapan, misalnya, setelah kode aplikasi diperiksa keluar dari repositori tetapi sebelum Apache dimulai ulang.

Chef menyebarkan aplikasi dalam empat tahap:

- Checkout —Mengunduh file dari repositori
- Migrasi —Menjalankan migrasi, seperti yang diperlukan
- Symlink —Membuat symlink
- Mulai ulang —Mulai ulang aplikasi

Kait penyebaran koki menyediakan cara sederhana untuk menyesuaikan penerapan dengan menjalankan aplikasi Ruby yang disediakan pengguna secara opsional setelah setiap tahap selesai. Untuk menggunakan kait penerapan, terapkan satu atau beberapa aplikasi Ruby dan letakkan di direktori aplikasi Anda. `/deploy` (Jika aplikasi Anda tidak memiliki `/deploy` direktori, buat direktori di `APP_ROOT` level tersebut.) Aplikasi harus memiliki salah satu nama berikut, yang menentukan kapan dijalankan.

- `before_migrate.rb`berjalan setelah tahap Checkout selesai tetapi sebelum Migrasi.
- `before_symlink.rb`berjalan setelah tahap Migrasi selesai tetapi sebelum Symlink.
- `before_restart.rb`berjalan setelah tahap Symlink selesai tetapi sebelum Restart.
- `after_restart.rb`berjalan setelah tahap Restart selesai.

Kait penerapan chef dapat mengakses objek node dengan menggunakan sintaks node standar, seperti resep. Deployment hooks juga dapat mengakses nilai variabel lingkungan aplikasi apa pun yang telah Anda tentukan. Namun, Anda harus menggunakan `new_resource.environment["`*`VARIABLE_NAME`*`"]` untuk mengakses nilai variabel alih-alih`ENV["`*`VARIABLE_NAME`*`"]`.

Menjalankan Cron Jobs di Instans Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pekerjaan cron Linux mengarahkan daemon cron untuk menjalankan satu atau lebih perintah pada jadwal yang ditentukan. Misalnya, tumpukan Anda mendukung aplikasi e-commerce PHP. Anda dapat mengatur pekerjaan cron agar server mengirimi Anda laporan penjualan pada waktu yang ditentukan setiap minggu. Untuk informasi lebih lanjut tentang cron, lihat cron di Wikipedia. Untuk informasi lebih lanjut tentang cara menjalankan pekerjaan cron langsung di komputer atau instance berbasis Linux, lihat Apa itu cron dan crontab, dan bagaimana cara menggunakannya? di situs web basis pengetahuan Universitas Indiana.

Meskipun Anda dapat secara manual mengatur `cron` pekerjaan pada instance berbasis Linux individu dengan menghubungkannya dengan SSH, dan mengedit `crontab` entri mereka, keuntungan utama dari AWS OpsWorks Stacks adalah Anda dapat mengarahkannya untuk menjalankan tugas di seluruh lapisan instance. Prosedur berikut menjelaskan cara mengatur `cron` pekerjaan pada instance lapisan PHP App Server, tetapi Anda dapat menggunakan pendekatan yang sama dengan lapisan apa pun.

Untuk menyiapkan **`cron`** pekerjaan pada instance layer

1. Terapkan buku masak dengan resep dengan `cron` sumber daya yang mengatur pekerjaan. Contoh mengasumsikan bahwa resep diberi nama`cronjob.rb`; rincian implementasi dijelaskan nanti. Untuk informasi lebih lanjut tentang buku masak dan resep, lihatBuku Masak dan Resep.
2. Instal buku masak di tumpukan Anda. Untuk informasi selengkapnya, lihat Memasang Buku Masak Kustom.
3. Minta AWS OpsWorks Stacks menjalankan resep secara otomatis pada instance layer dengan menetapkannya ke peristiwa siklus hidup berikut. Untuk informasi selengkapnya, lihat Resep Menjalankan Secara Otomatis.

    - Pengaturan - Menetapkan `cronjob.rb` ke acara ini mengarahkan AWS OpsWorks Stacks untuk menjalankan resep pada semua instance baru.
    - Deploy — Menetapkan `cronjob.rb` ke acara ini mengarahkan AWS OpsWorks Stacks untuk menjalankan resep di semua instance online saat Anda menerapkan atau menerapkan ulang aplikasi ke layer.

Anda juga dapat menjalankan resep secara manual pada instance online dengan menggunakan perintah `Execute Recipes` stack. Untuk informasi selengkapnya, lihat Perintah Tumpukan.

Berikut ini adalah `cronjob.rb` contoh, yang menyiapkan pekerjaan cron untuk menjalankan aplikasi PHP yang diimplementasikan pengguna seminggu sekali yang mengumpulkan data penjualan dari server dan mengirimkan laporan. Untuk contoh lebih lanjut tentang cara menggunakan sumber daya cron, lihat cron.

```
cron "job_name" do
  hour "1"
  minute "10"
  weekday "6"
  command "cd /srv/www/myapp/current && php .lib/mailing.php"
end
```

`cron`adalah sumber daya Chef yang mewakili `cron` pekerjaan. Saat AWS OpsWorks Stacks menjalankan resep pada sebuah instance, penyedia terkait menangani detail pengaturan pekerjaan.

- *`job_name`*adalah nama yang ditentukan pengguna untuk `cron` pekerjaan itu, seperti. `weekly report`
- `hour`/`minute`/`weekday`tentukan kapan perintah harus dijalankan. Contoh ini menjalankan perintah setiap hari Sabtu pukul 1:10 pagi.
- `command`menentukan perintah yang akan dijalankan.

  Contoh ini menjalankan dua perintah. Yang pertama menavigasi ke `/srv/www/myapp/current` direktori. Yang kedua menjalankan `mailing.php` aplikasi yang diterapkan pengguna, yang mengumpulkan data penjualan dan mengirimkan laporan.

> **Note**
>
> `bundle`Perintah tidak bekerja dengan `cron` pekerjaan secara default. Alasannya adalah bahwa AWS OpsWorks Stacks menginstal bundler di direktori. `/usr/local/bin` Untuk digunakan `bundle` dengan `cron` pekerjaan, Anda harus secara eksplisit menambahkan jalur `/usr/local/bin` ke pekerjaan cron. Juga, karena variabel lingkungan $PATH mungkin tidak berkembang dalam `cron` pekerjaan, praktik terbaik adalah secara eksplisit

menambahkan informasi jalur yang diperlukan ke pekerjaan tanpa bergantung pada perluasan variabel $PATH. Contoh berikut menunjukkan dua cara untuk digunakan `bundle` dalam suatu `cron` pekerjaan.

```
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
```

Jika tumpukan Anda memiliki beberapa server aplikasi, menetapkan `cronjob.rb` ke peristiwa siklus hidup lapisan PHP App Server mungkin bukan pendekatan yang ideal. Misalnya, resep berjalan pada semua instance layer, sehingga Anda akan menerima beberapa laporan. Pendekatan yang lebih baik adalah menggunakan lapisan khusus untuk memastikan bahwa hanya satu server yang mengirim laporan.

Untuk menjalankan resep hanya pada salah satu instance layer

1. Buat layer khusus yang disebut, misalnya, PHPAdmin dan tetapkan `cronjob.rb` ke acara Setup dan Deploy. Lapisan khusus tidak harus melakukan banyak hal. Dalam hal ini, jalankan PHPAdmin saja satu resep khusus pada instance-nya.
2. Tetapkan salah satu instance PHP App Server ke. AdminLayer Jika sebuah instance milik lebih dari satu lapisan, AWS OpsWorks Stacks menjalankan resep bawaan dan kustom setiap lapisan.

Karena hanya satu instance milik PHP App Server dan PHPAdmin layer, `cronjob.rb` berjalan hanya pada instance itu dan Anda hanya menerima satu laporan.

Menginstal dan Mengkonfigurasi Paket di Instans Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Lapisan bawaan hanya mendukung paket tertentu. Untuk informasi selengkapnya, lihat Lapisan. Anda dapat menginstal paket lain, seperti server Redis, dengan menerapkan resep khusus untuk menangani tugas penyiapan, konfigurasi, dan penerapan terkait. Dalam beberapa kasus, pendekatan terbaik adalah memperluas lapisan bawaan agar menginstal paket pada instance-nya di samping paket standar layer. Misalnya, jika Anda memiliki tumpukan yang mendukung aplikasi PHP, dan Anda ingin menyertakan server Redis, Anda dapat memperluas lapisan PHP App Server untuk menginstal dan mengkonfigurasi server Redis pada instance lapisan selain server aplikasi PHP.

Resep instalasi paket biasanya perlu melakukan tugas-tugas seperti ini:

- Buat satu atau lebih direktori dan atur modenya.
- Buat file konfigurasi dari template.
- Jalankan installer untuk menginstal paket pada instance.
- Mulai satu atau lebih layanan.

Untuk contoh cara menginstal server Tomcat, lihatMembuat Layer Server Tomcat Kustom. Topik menjelaskan cara mengatur lapisan Redis kustom, tetapi Anda dapat menggunakan banyak kode yang sama untuk menginstal dan mengkonfigurasi Redis pada lapisan bawaan. Untuk contoh cara menginstal paket lain, lihat buku masak bawaan, di https://github.com/aws/opsworks-cookbooks.

Membuat Layer Server Tomcat Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Topik ini menjelaskan cara menerapkan lapisan khusus untuk tumpukan Linux. Namun, prinsip dasar dan beberapa kode juga dapat disesuaikan untuk mengimplementasikan lapisan khusus untuk tumpukan Windows, terutama yang ada di bagian penerapan aplikasi.

Cara termudah untuk menggunakan paket non-standar pada instance AWS OpsWorks Stacks adalah dengan memperluas lapisan yang ada. Namun, pendekatan ini menginstal dan menjalankan paket standar dan tidak standar pada instance lapisan, yang tidak selalu diinginkan. Pendekatan yang agak lebih menuntut tetapi lebih kuat adalah menerapkan lapisan khusus, yang memberi Anda kontrol hampir penuh atas instance lapisan, termasuk yang berikut ini:

- Paket mana yang diinstal
- Bagaimana setiap paket dikonfigurasi
- Cara menyebarkan aplikasi dari repositori ke instance

Baik menggunakan konsol atau API, Anda membuat dan mengelola lapisan khusus seperti lapisan lainnya, seperti yang dijelaskan dalamLapisan Kustom. Namun, resep bawaan lapisan khusus hanya melakukan beberapa tugas yang sangat mendasar, seperti menginstal klien Ganglia untuk melaporkan metrik ke master Ganglia. Untuk membuat instance lapisan kustom lebih dari fungsional minimal, Anda harus menerapkan satu atau lebih buku masak khusus dengan resep Chef dan file terkait untuk menangani tugas menginstal dan mengonfigurasi paket, menerapkan aplikasi, dan sebagainya. Anda tidak perlu mengimplementasikan semuanya dari awal. Misalnya, jika Anda menyimpan aplikasi di salah satu repositori standar, Anda dapat menggunakan resep penerapan bawaan untuk menangani banyak pekerjaan menginstal aplikasi pada instance lapisan.

> **Note**
>
> Jika Anda baru mengenal Chef, Anda harus membaca terlebih dahuluBuku masak 101, yang merupakan tutorial yang memperkenalkan dasar-dasar bagaimana menerapkan buku masak untuk melakukan berbagai tugas umum.

Panduan berikut menjelaskan cara menerapkan lapisan khusus yang mendukung server aplikasi Tomcat. Lapisan ini didasarkan pada buku masak khusus bernama Tomcat, yang mencakup resep untuk menangani instalasi paket, penyebaran, dan sebagainya. Panduan ini mencakup kutipan dari buku masak Tomcat. Anda dapat mengunduh buku masak lengkap dari GitHub repositorinya. Jika Anda tidak terbiasa dengan Opscode Chef, Anda harus membaca terlebih dahulu. Buku Masak dan Resep

> **Note**
>
> AWS OpsWorks Stacks menyertakan lapisan Java App Server berfitur lengkap untuk penggunaan produksi. Tujuan dari buku masak Tomcat adalah untuk menunjukkan bagaimana menerapkan lapisan khusus, sehingga hanya mendukung versi terbatas Tomcat yang tidak menyertakan fitur seperti SSL. Untuk contoh implementasi berfitur lengkap, lihat buku masak opsworks_java bawaan.

Buku masak Tomcat mendukung lapisan khusus yang instancenya memiliki karakteristik sebagai berikut:

- Mereka mendukung server aplikasi Tomcat Java dengan front end Apache.
- Tomcat dikonfigurasi untuk memungkinkan aplikasi menggunakan `DataSource` objek JDBC untuk terhubung ke instance MySQL terpisah, yang berfungsi sebagai penyimpanan data back end.

Buku masak untuk proyek ini melibatkan beberapa komponen utama:

- File atribut berisi pengaturan konfigurasi yang digunakan oleh berbagai resep.
- Resep penyiapan ditetapkan ke acara siklus hidup Pengaturan lapisan. Mereka berjalan setelah sebuah instance telah boot dan melakukan tugas-tugas seperti menginstal paket dan membuat file konfigurasi.
- Konfigurasi resep ditetapkan ke acara Configure lifecycle layer. Mereka berjalan setelah perubahan konfigurasi tumpukan — terutama ketika instance online atau offline — dan menangani perubahan konfigurasi yang diperlukan.
- Resep deploy ditetapkan ke acara siklus hidup Deploy layer. Mereka berjalan setelah resep Pengaturan dan saat Anda menerapkan aplikasi secara manual untuk menginstal kode dan file terkait pada instance lapisan dan menangani tugas terkait, seperti memulai ulang layanan.

Bagian terakhirBuat Stack dan Jalankan Aplikasi, menjelaskan cara membuat tumpukan yang mencakup lapisan khusus berdasarkan buku masak Tomcat dan cara menyebarkan dan menjalankan aplikasi JSP sederhana yang menampilkan data dari database MySQL yang berjalan pada instance milik lapisan MySQL yang terpisah.

> **Note**
>
> Resep buku masak Tomcat bergantung pada beberapa resep bawaan AWS OpsWorks Stacks. Untuk memperjelas asal setiap resep, topik ini mengidentifikasi resep menggunakan buku masak Chef:: konvensi recipename.

Topik

- Berkas Atribut
- Resep Pengaturan
- Konfigurasikan Resep
- Menyebarkan Resep
- Buat Stack dan Jalankan Aplikasi

Berkas Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sebelum melihat resep, penting untuk terlebih dahulu memeriksa file atribut buku masak Tomcat, yang berisi berbagai pengaturan konfigurasi yang digunakan resep. Atribut tidak diperlukan; Anda cukup hardcode nilai-nilai ini dalam resep atau template Anda. Namun, jika Anda menentukan pengaturan konfigurasi menggunakan atribut, Anda dapat menggunakan konsol AWS OpsWorks Stacks atau API untuk memodifikasi nilai dengan mendefinisikan atribut JSON kustom, yang lebih sederhana dan lebih fleksibel daripada menulis ulang resep atau kode templat setiap kali Anda ingin

mengubah pengaturan. Pendekatan ini memungkinkan Anda, misalnya, untuk menggunakan buku masak yang sama untuk beberapa tumpukan, tetapi mengkonfigurasi server Tomcat secara berbeda untuk setiap tumpukan. Untuk informasi selengkapnya tentang atribut dan cara menggantinya, lihatMengesampingkan Atribut.

Contoh berikut menunjukkan file atribut lengkap,`default.rb`, yang terletak di direktori buku masak Tomcat. `attributes`

```
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
```

Pengaturan itu sendiri dibahas nanti di bagian terkait. Catatan berikut berlaku secara umum:

- Semua definisi node adalah `default` tipe, sehingga Anda dapat menggantikannya dengan atribut JSON khusus.
- File ini menggunakan `case` pernyataan untuk mengatur beberapa nilai atribut secara kondisional berdasarkan sistem operasi instance.

  `platform`Node dihasilkan oleh alat Ohai Chef dan mewakili sistem operasi instance.

Resep Pengaturan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Resep penyiapan ditetapkan ke acara siklus hidup Pengaturan lapisan dan dijalankan setelah instance boot. Mereka melakukan tugas-tugas seperti menginstal paket, membuat file konfigurasi, dan memulai layanan. Setelah resep Pengaturan selesai berjalan, AWS OpsWorks Stacks menjalankan resep Deploy untuk menerapkan aplikasi apa pun ke instance baru.

Topik

- tomcat: :pengaturan
- tomcat: :instal
- kucing jantan: :layanan
- kucing jantan: :container_config
- kucing jantan: :apache_tomcat_bind

tomcat: :pengaturan

`tomcat::setup`Resep ini dimaksudkan untuk ditetapkan ke acara siklus hidup Pengaturan lapisan.

```
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'
```

`tomcat::setup`presep sebagian besar merupakan metarecipe. Ini termasuk satu set resep dependen yang menangani sebagian besar rincian menginstal dan mengkonfigurasi Tomcat dan paket terkait. Bagian pertama `tomcat::setup` menjalankan resep berikut, yang dibahas nanti:

- tomcat: :instalResepnya menginstal paket server Tomcat.
- kucing jantan: :layananResepnya mengatur layanan Tomcat.

Bagian tengah `tomcat::setup` mengaktifkan dan memulai layanan Tomcat:

- Sumber daya layanan Chef memungkinkan layanan Tomcat saat boot.
- Sumber daya Chef bash menjalankan skrip Bash untuk memulai daemon otomatis, yang diperlukan untuk instance yang didukung Amazon EBS. Sumber daya kemudian memberi tahu `service` sumber daya untuk memulai ulang layanan Tomcat.

  Untuk informasi selengkapnya, lihat: autofs (untuk Amazon Linux) atau Autofs (untuk Ubuntu).

Bagian terakhir dari `tomcat::setup` membuat file konfigurasi dan menginstal dan mengkonfigurasi server Apache front-end:

- kucing jantan: :container_configResep membuat file konfigurasi.
- `apache2`Resep (yang merupakan singkatan dari`apache2::default`) adalah resep bawaan AWS OpsWorks Stacks yang menginstal dan mengonfigurasi server Apache.
- kucing jantan: :apache_tomcat_bindResep mengkonfigurasi server Apache untuk berfungsi sebagai front-end untuk server Tomcat.

> **Note**
>
> Anda sering dapat menghemat waktu dan tenaga dengan menggunakan resep bawaan untuk melakukan beberapa tugas yang diperlukan. Resep ini menggunakan `apache2::default` resep bawaan untuk menginstal Apache daripada menerapkannya dari awal. Untuk contoh lain tentang cara menggunakan resep bawaan, lihatMenyebarkan Resep.

Bagian berikut menjelaskan resep Pengaturan buku masak Tomcat secara lebih rinci. Untuk informasi lebih lanjut tentang `apache2` resep, lihat opsworks-cookbooks/apache2.

tomcat: :instal

`tomcat::install` Resep menginstal server Tomcat, OpenJDK, dan pustaka konektor Java yang menangani koneksi ke server MySQL.

```
tomcat_pkgs = value_for_platform(
  ['debian', 'ubuntu'] => {
    'default' => ["tomcat#{node['tomcat']['base_version']}", 'libtcnative-1',
 'libmysql-java']
  },
  ['centos', 'redhat', 'fedora', 'amazon'] => {
    'default' => ["tomcat#{node['tomcat']['base_version']}", 'tomcat-native', 'mysql-
connector-java']
  },
  'default' => ["tomcat#{node['tomcat']['base_version']}"]
)

tomcat_pkgs.each do |pkg|
  package pkg do
    action :install
  end
```

```
end

link ::File.join(node['tomcat']['lib_dir'], node['tomcat']['mysql_connector_jar']) do
  to ::File.join(node['tomcat']['java_dir'], node['tomcat']['mysql_connector_jar'])
  action :create
end

# remove the ROOT webapp, if it got installed by default
include_recipe 'tomcat::remove_root_webapp'
```

Resep melakukan tugas-tugas berikut:

1. Membuat daftar paket yang akan diinstal, tergantung pada sistem operasi instans.
2. Menginstal setiap paket dalam daftar.

   Sumber daya paket Chef menggunakan penyedia yang sesuai — `yum` untuk Amazon Linux dan `apt-get` untuk Ubuntu — untuk menangani instalasi. Penyedia paket menginstal OpenJDK sebagai ketergantungan Tomcat, tetapi pustaka konektor MySQL harus diinstal secara eksplisit.
3. Menggunakan sumber daya tautan Chef untuk membuat symlink di direktori lib server Tomcat ke pustaka konektor MySQL di JDK.

   Menggunakan nilai atribut default, direktori lib Tomcat adalah `/usr/share/tomcat6/lib` dan perpustakaan konektor MySQL () ada di. `mysql-connector-java.jar` `/usr/share/java/`

`tomcat::remove_root_webapp`Resep menghapus aplikasi web ROOT (secara `/var/lib/tomcat6/webapps/ROOT` default) untuk menghindari beberapa masalah keamanan.

```
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
```

`only_if`Pernyataan tersebut memastikan bahwa resep menghapus file hanya jika ada.

> **Note**
>
> Versi Tomcat ditentukan oleh `['tomcat']['base_version']` atribut, yang diatur ke 6 dalam file atribut. Untuk menginstal Tomcat 7, Anda dapat menggunakan atribut JSON khusus untuk mengganti atribut. Cukup edit pengaturan tumpukan Anda dan masukkan JSON berikut di kotak Custom Chef JSON, atau tambahkan ke JSON kustom yang ada:
>
> ```
> {
>   'tomcat' : {
>     'base_version' : 7
>   }
> }
> ```
>
> Atribut JSON kustom mengganti atribut default dan menetapkan versi Tomcat ke 7. Untuk informasi selengkapnya tentang atribut utama, lihatMengesampingkan Atribut.

kucing jantan: :layanan

`tomcat::service`Resepnya menciptakan definisi layanan 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
```

Resep menggunakan sumber daya layanan Chef untuk menentukan nama layanan Tomcat (tomcat6, secara default) dan menetapkan `supports` atribut untuk menentukan bagaimana Chef mengelola perintah restart, reload, dan status layanan pada sistem operasi yang berbeda.

- `true`menunjukkan bahwa Chef dapat menggunakan skrip init atau penyedia layanan lain untuk menjalankan perintah
- `false`menunjukkan bahwa Chef harus mencoba menjalankan perintah dengan cara lain.

Perhatikan bahwa `action` diatur ke`:nothing`. Untuk setiap peristiwa siklus hidup, AWS OpsWorks Stacks memulai proses Chef untuk mengeksekusi set resep yang sesuai. Buku masak Tomcat mengikuti pola umum memiliki resep yang membuat definisi layanan, tetapi tidak memulai ulang layanan. Resep lain di Chef run menangani restart, biasanya dengan memasukkan `notifies` perintah dalam `template` sumber daya yang digunakan untuk membuat file konfigurasi. Pemberitahuan adalah cara mudah untuk memulai ulang layanan karena mereka melakukannya hanya jika konfigurasi telah berubah. Selain itu, jika Chef run memiliki beberapa notifikasi restart untuk suatu layanan, Chef memulai ulang layanan paling banyak sekali. Praktik ini menghindari masalah yang dapat terjadi ketika mencoba memulai kembali layanan yang tidak sepenuhnya beroperasi, yang merupakan sumber umum kesalahan Tomcat.

Layanan Tomcat harus ditentukan untuk setiap Chef run yang menggunakan notifikasi restart. `tomcat::service`Oleh karena itu termasuk dalam beberapa resep, untuk memastikan bahwa layanan ditentukan untuk setiap Chef menjalankan. Tidak ada penalti jika Chef run menyertakan beberapa contoh `tomcat::service` karena Chef memastikan bahwa resep dijalankan hanya sekali per run, terlepas dari berapa kali itu disertakan.

kucing jantan: :container_config

`tomcat::container_config`Resep membuat file konfigurasi dari file template buku masak.

```
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
```

Resep panggilan pertama`tomcat::service`, yang mendefinisikan layanan jika perlu. Sebagian besar resep terdiri dari dua sumber template, yang masing-masing membuat file konfigurasi dari salah satu file template buku masak, menetapkan properti file, dan memberi tahu Chef untuk memulai ulang layanan.

File Konfigurasi Lingkungan Tomcat

`template`Sumber daya pertama menggunakan file `tomcat_env_config.erb` template untuk membuat file konfigurasi lingkungan Tomcat, yang digunakan untuk mengatur variabel lingkungan seperti`JAVA_HOME`. Nama file default adalah argumen `template` sumber daya. `tomcat::container_config`menggunakan `path` atribut untuk mengganti nilai default dan memberi nama file konfigurasi `/etc/sysconfig/tomcat6` (Amazon Linux) atau `/etc/default/tomcat6` (Ubuntu). `template`Sumber daya juga menentukan pemilik file, grup, dan pengaturan mode dan mengarahkan Chef untuk tidak membuat file cadangan.

Jika Anda melihat kode sumber, sebenarnya ada tiga versi`tomcat_env_config.erb`, masing-masing dalam subdirektori yang berbeda dari `templates` direktori. `amazon`Direktori `ubuntu` dan berisi template untuk sistem operasi masing-masing. `default`Folder berisi template dummy dengan satu baris komentar, yang hanya digunakan jika Anda mencoba menjalankan resep ini pada instance dengan sistem operasi yang tidak didukung. `tomcat::container_config`Resepnya tidak perlu menentukan mana yang `tomcat_env_config.erb` akan digunakan. Chef secara otomatis memilih direktori yang sesuai untuk sistem operasi instans berdasarkan aturan yang dijelaskan dalam Kekhususan File.

`tomcat_env_config.erb`File untuk contoh ini sebagian besar terdiri dari komentar. Untuk mengatur variabel lingkungan tambahan, hapus komentar pada baris yang sesuai dan berikan nilai pilihan Anda.

> **Note**
>
> Pengaturan konfigurasi apa pun yang mungkin berubah harus didefinisikan sebagai atribut daripada di-hardcode dalam template. Dengan begitu, Anda tidak perlu menulis ulang template untuk mengubah pengaturan, Anda cukup mengganti atribut.

Template Amazon Linux hanya menetapkan satu variabel lingkungan, seperti yang ditunjukkan dalam kutipan berikut.

```
...
# 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 dapat digunakan untuk menentukan opsi Java seperti jalur perpustakaan. Menggunakan nilai atribut default, template tidak menetapkan opsi Java untuk Amazon Linux. Anda dapat mengatur opsi Java Anda sendiri dengan mengganti `['tomcat']['java_opts']` atribut, misalnya, dengan menggunakan atribut JSON kustom. Sebagai contoh, lihat Buat Stack.

Template Ubuntu menetapkan beberapa variabel lingkungan, seperti yang ditunjukkan dalam kutipan template berikut.

```
# 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 -%>
```

Menggunakan nilai atribut default, template menetapkan variabel lingkungan Ubuntu sebagai berikut:

- `TOMCAT6_USER`dan`TOMCAT6_GROUP`, yang mewakili pengguna dan grup Tomcat, keduanya disetel `ketomcat6`.

  Jika Anda menyetel ['tomcat'] ['base_version'] `ketomcat7`, nama variabel diselesaikan ke `TOMCAT7_USER` dan`TOMCAT7_GROUP`, dan keduanya disetel ke. `tomcat7`
- `JAVA_OPTS`diatur ke`-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`:
  - Pengaturan `-Djava.awt.headless` untuk `true` menginformasikan mesin grafis bahwa instance tanpa kepala dan tidak memiliki konsol, yang membahas perilaku yang salah dari aplikasi grafis tertentu.
  - `-Xmx128m`memastikan bahwa JVM memiliki sumber daya memori yang memadai, 128MB untuk contoh ini.
  - `-XX:+UseConcMarkSweepGC`menentukan pengumpulan sampah sapuan tanda bersamaan, yang membantu membatasi jeda yang disebabkan oleh pengumpulan sampah.

    Untuk informasi lebih lanjut, lihat: Penyempurnaan Kolektor Sapu Tanda Bersamaan.
- Jika versi Tomcat kurang dari 7, template tidak disetel`LC_ALL`, yang membahas bug Ubuntu.

> **Note**
>
> Dengan atribut default, beberapa variabel lingkungan ini hanya diatur ke nilai default mereka. Namun, secara eksplisit menyetel variabel lingkungan ke atribut berarti Anda dapat menentukan atribut JSON khusus untuk mengganti atribut default dan memberikan nilai khusus. Untuk informasi selengkapnya tentang atribut utama, lihatMengesampingkan Atribut.

Untuk file template lengkap, lihat kode sumber.

File Konfigurasi Server.xml

`template`Sumber daya kedua digunakan `server.xml.erb` untuk membuat file `system.xml` konfigurasi, yang mengkonfigurasi servlet/JSP wadah. `server.xml.erb`tidak berisi pengaturan khusus sistem operasi, sehingga ada di subdirektori `template` direktori. `default`

Template menggunakan pengaturan standar, tetapi dapat membuat `system.xml` file untuk Tomcat 6 atau Tomcat 7. Misalnya, kode berikut dari bagian server template mengonfigurasi pendengar dengan tepat untuk versi yang ditentukan.

```
<% 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 -%>
```

Template menggunakan atribut sebagai pengganti pengaturan hardcode sehingga Anda dapat dengan mudah mengubah pengaturan dengan mendefinisikan atribut JSON kustom. Misalnya:

```
<Connector port="<%= node['tomcat']['port'] %>" protocol="HTTP/1.1"
           connectionTimeout="20000"
           URIEncoding="<%= node['tomcat']['uri_encoding'] %>"
           redirectPort="<%= node['tomcat']['secure_port'] %>" />
```

Untuk informasi selengkapnya, lihat kode sumber.

kucing jantan: :apache_tomcat_bind

`tomcat::apache_tomcat_bind`Resepnya memungkinkan server Apache untuk bertindak sebagai front end Tomcat, menerima permintaan masuk dan meneruskannya ke Tomcat dan mengembalikan tanggapan ke klien. Contoh ini menggunakan mod_proxy sebagai proxy/gateway Apache.

```
execute 'enable mod_proxy for apache-tomcat binding' do
  command '/usr/sbin/a2enmod proxy'
  not_if do
    ::File.symlink?(::File.join(node['apache']['dir'], 'mods-enabled', 'proxy.load'))
 || node['tomcat']['apache_tomcat_bind_mod'] !~ /\Aproxy/
  end
end

execute 'enable module for apache-tomcat binding' do
  command "/usr/sbin/a2enmod #{node['tomcat']['apache_tomcat_bind_mod']}"
  not_if {::File.symlink?(::File.join(node['apache']['dir'], 'mods-enabled',
 "#{node['tomcat']['apache_tomcat_bind_mod']}.load"))}
end

include_recipe 'apache2::service'

template 'tomcat thru apache binding' do
  path ::File.join(node['apache']['dir'], 'conf.d', node['tomcat']
['apache_tomcat_bind_config'])
  source 'apache_tomcat_bind.conf.erb'
  owner 'root'
  group 'root'
  mode 0644
  backup false
  notifies :restart, resources(:service => 'apache2')
end
```

Untuk mengaktifkanmod_proxy, Anda harus mengaktifkan `proxy` modul dan modul berbasis protokol. Anda memiliki dua opsi untuk modul protokol:

- HTTP: `proxy_http`
- JServ Protokol Apache (AJP): `proxy_ajp`

  AJP adalah protokol Tomcat internal.

Kedua sumber daya eksekusi resep menjalankan `a2enmod` perintah, yang memungkinkan modul yang ditentukan dengan membuat symlink yang diperlukan:

- `execute`Sumber daya pertama memungkinkan `proxy` modul.
- `execute`Sumber daya kedua memungkinkan modul protokol, yang diatur ke secara `proxy_http` default.

  Jika Anda lebih suka menggunakan AJP, Anda dapat menentukan JSON khusus untuk mengganti `apache_tomcat_bind_mod` atribut dan mengaturnya ke. `proxy_ajp`

`apache2::service`Resepnya adalah resep bawaan AWS OpsWorks Stacks yang mendefinisikan layanan Apache. Untuk informasi lebih lanjut, lihat resep di GitHub repositori AWS OpsWorks Stacks.

Sumber `template` daya digunakan `apache_tomcat_bind.conf.erb` untuk membuat file konfigurasi, dinamai secara `tomcat_bind.conf` default. Ini menempatkan file di `['apache']['dir']/.conf.d` direktori. `['apache']['dir']`Atribut didefinisikan dalam file `apache2` atribut bawaan, dan diatur secara default ke `/etc/httpd` (Amazon Linux), atau `/etc/apache2` (Ubuntu). Jika `template` sumber daya membuat atau mengubah file konfigurasi, `notifies` perintah menjadwalkan restart layanan 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 -%>
```

Template menggunakan ProxyPassReversearahan ProxyPassdan untuk mengkonfigurasi port yang digunakan untuk melewati lalu lintas antara Apache dan Tomcat. Karena kedua server berada pada instance yang sama, mereka dapat menggunakan URL localhost dan keduanya diatur secara default ke. `http://localhost:8080`

Konfigurasikan Resep

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Konfigurasi resep ditetapkan ke peristiwa siklus hidup Konfigurasi lapisan, yang terjadi pada semua instance tumpukan setiap kali instance masuk atau keluar dari status online. Anda menggunakan Configure recipes untuk menyesuaikan konfigurasi instans untuk merespons perubahan, sebagaimana mestinya. Saat Anda menerapkan resep Konfigurasi, perlu diingat bahwa perubahan konfigurasi tumpukan mungkin melibatkan instance yang tidak ada hubungannya dengan lapisan ini. Resep harus dapat merespons dengan tepat, yang mungkin berarti tidak melakukan apa-apa dalam beberapa kasus.

tomcat: :konfigurasikan

`tomcat::configure`Resep ini ditujukan untuk acara siklus hidup Konfigurasi lapisan.

```
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'
```

`tomcat::configure`Resepnya pada dasarnya adalah metarecipe yang menjalankan dua resep tergantung.

1. `tomcat::context`Resep membuat file konfigurasi konteks aplikasi web.

   File ini mengkonfigurasi sumber daya JDBC yang digunakan aplikasi untuk berkomunikasi dengan instance MySQL, seperti yang dibahas di bagian berikutnya. Menjalankan resep ini sebagai respons terhadap peristiwa konfigurasi memungkinkan layer untuk memperbarui file konfigurasi konteks aplikasi web jika lapisan database telah berubah.

2. Resep `tomcat::container_config` Pengaturan dijalankan lagi untuk menangkap perubahan apa pun dalam konfigurasi kontainer.

The `include` for `tomcat::container_config` dikomentari untuk contoh ini. Jika Anda ingin menggunakan JSON khusus untuk memodifikasi pengaturan Tomcat, Anda dapat menghapus komentar. Peristiwa siklus hidup Konfigurasi kemudian berjalan`tomcat::container_config`, yang memperbarui file konfigurasi terkait Tomcat, seperti yang dijelaskan dalam kucing jantan: :container_config dan memulai ulang layanan Tomcat.

kucing jantan: :konteks

Cookbook Tomcat memungkinkan aplikasi untuk mengakses server database MySQL, yang dapat berjalan pada instance terpisah, dengan menggunakan objek J2EE. DataSource Dengan Tomcat, Anda dapat mengaktifkan koneksi dengan membuat dan menginstal file konfigurasi konteks aplikasi web untuk setiap aplikasi. File ini mendefinisikan hubungan antara aplikasi dan sumber daya JDBC yang akan digunakan aplikasi untuk berkomunikasi dengan database. Untuk informasi selengkapnya, lihat Context Container.

Tujuan utama `tomcat::context` resep adalah untuk membuat file konfigurasi ini.

```
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
```

Selain atribut buku masak Tomcat, resep ini menggunakan konfigurasi tumpukan dan atribut penerapan yang diinstal AWS OpsWorks Stacks dengan peristiwa Configure. Layanan AWS OpsWorks Stacks menambahkan atribut ke objek node setiap instance yang berisi informasi yang biasanya diperoleh resep dengan menggunakan tas data atau mencari dan menginstal atribut pada setiap instance. Atribut berisi informasi terperinci tentang konfigurasi tumpukan, aplikasi yang diterapkan, dan data kustom apa pun yang ingin disertakan pengguna. Resep dapat memperoleh data dari konfigurasi tumpukan dan atribut penerapan dengan menggunakan sintaks simpul Chef standar. Untuk informasi selengkapnya, lihat Konfigurasi Stack dan Atribut Deployment. Dengan tumpukan Chef 11.10, Anda juga dapat menggunakan pencarian Chef untuk mendapatkan konfigurasi tumpukan dan data penyebaran. Untuk informasi selengkapnya, lihat Menggunakan Pencarian Chef.

`deploy`atribut mengacu pada `[:deploy]` namespace, yang berisi atribut terkait penerapan yang ditentukan melalui konsol atau API, atau dihasilkan oleh layanan Stacks. AWS OpsWorks `deploy`Atribut tersebut menyertakan atribut untuk setiap aplikasi yang di-deploy, dinamai dengan nama pendek aplikasi. Setiap atribut app berisi sekumpulan atribut yang menjadi ciri aplikasi, seperti root dokumen (`[:deploy][:`*`appname`*`][:document_root]`).

`context`Resep pertama memastikan bahwa layanan ditentukan untuk Chef ini dijalankan dengan meneleponkucing jantan: :layanan. Kemudian mendefinisikan `context_name` variabel yang mewakili nama file konfigurasi, tidak termasuk ekstensi. `.xml` Jika Anda menggunakan root dokumen default, `context_name` disetel ke nama pendek aplikasi. Jika tidak, itu diatur ke root dokumen yang ditentukan. Contoh yang dibahas dalam Buat Stack dan Jalankan Aplikasi menetapkan root dokumen`"ROOT"`, sehingga konteksnya adalah ROOT dan file konfigurasi diberi nama`ROOT.xml`.

Sebagian besar resep melewati daftar aplikasi yang diterapkan dan untuk setiap aplikasi, menggunakan `webapp_context.xml.erb` template untuk membuat file konfigurasi konteks. Contoh ini hanya menggunakan satu aplikasi, tetapi definisi `deploy` atribut mengharuskan Anda untuk memperlakukannya sebagai daftar aplikasi apa pun.

`webapp_context.xml.erb`Template tidak spesifik sistem operasi, sehingga terletak di `templates` subdirektori direktori. `default`

Resep membuat file konfigurasi sebagai berikut:

- Menggunakan nilai atribut default, nama file konfigurasi diatur ke *`context_name`*`.xml` dan diinstal di `/etc/tomcat6/Catalina/localhost/` direktori.

`['datasources']`Node dari atribut konfigurasi tumpukan berisi satu atau lebih atribut, yang masing-masing memetakan nama konteks ke sumber daya data JDBC yang akan digunakan aplikasi terkait untuk berkomunikasi dengan database. Node dan isinya didefinisikan dengan JSON khusus saat Anda membuat tumpukan, seperti yang dijelaskan nanti. Buat Stack dan Jalankan Aplikasi Contoh ini memiliki atribut tunggal yang mengaitkan nama konteks ROOT dengan sumber daya JDBC bernama jdbc/mydb.

- Menggunakan nilai atribut default, pengguna dan grup file keduanya diatur ke nilai yang ditentukan oleh paket Tomcat: `tomcat` (Amazon Linux) atau `tomcat6` (Ubuntu).
- `template`Sumber daya membuat file konfigurasi hanya jika `['datasources']` node ada dan menyertakan `context_name` atribut.
- Sumber `template` daya mendefinisikan dua variabel, `resource_name` dan`webapp_name`.

  `resource_name`disetel ke nama sumber daya yang dikaitkan dengan `context_name` dan `webapp_name` disetel ke nama pendek aplikasi.
- Sumber daya template memulai ulang layanan Tomcat untuk memuat dan mengaktifkan perubahan.

`webapp_context.xml.erb`Template terdiri dari `Context` elemen yang berisi `Resource` elemen dengan set atributnya sendiri.

`Resource`Atribut mencirikan konfigurasi konteks:

- nama —Nama sumber daya JDBC, yang diatur ke `resource_name` nilai yang ditentukan dalam. `tomcat::context`

  Sebagai contoh, nama sumber daya diatur ke jdbc/mydb.
- auth dan ketik —Ini adalah pengaturan standar untuk koneksi `DataSource` JDBC.
- MaxActive, MaxIdle, dan MaxWait —Jumlah maksimum koneksi aktif dan idle, dan waktu tunggu maksimum untuk koneksi dikembalikan.
- nama pengguna, dan kata sandi —Nama pengguna database dan kata sandi root, yang diperoleh dari `deploy` atribut.
- driverClassName—Nama kelas driver JDBC, yang diatur ke driver MySQL.
- url —URL koneksi.

Awalan tergantung pada database. Ini harus diatur ke `jdbc:mysql` untuk MySQL, untuk Postgres`jdbc:postgresql`, `jdbc:sqlserver` dan untuk SQL Server. Contoh menyetel URL `kejdbc:mysql://`*`host_IP_Address`*`:3306:simplejsp`, di *`simplejsp`* mana nama pendek aplikasi.

- pabrik `DataSource` —Pabrik, yang diperlukan untuk database MySQL.

Untuk informasi selengkapnya tentang file konfigurasi ini, lihat DataSources topik Menggunakan wiki Tomcat.

Menyebarkan Resep

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Resep deploy ditetapkan ke acara siklus hidup Deploy layer. Ini biasanya terjadi pada semua instance tumpukan setiap kali Anda menerapkan aplikasi, meskipun Anda secara opsional dapat membatasi acara hanya untuk instance tertentu. AWS OpsWorks Stacks juga menjalankan resep Deploy pada instance baru, setelah resep Setup selesai. Tujuan utama dari resep Deploy adalah untuk menyebarkan kode dan file terkait dari repositori ke instance lapisan server aplikasi. Namun, Anda sering menjalankan resep Deploy di lapisan lain juga. Hal ini memungkinkan instance lapisan tersebut, misalnya, memperbarui konfigurasi mereka untuk mengakomodasi aplikasi yang baru diterapkan. Saat Anda menerapkan resep Deploy, ingatlah bahwa peristiwa Deploy tidak selalu berarti bahwa aplikasi sedang di-deploy ke instance. Ini bisa berupa pemberitahuan bahwa aplikasi sedang digunakan ke instance lain di tumpukan, untuk memungkinkan instance melakukan pembaruan yang diperlukan. Resepnya harus dapat merespons dengan tepat, yang mungkin berarti tidak melakukan apa-apa.

AWS OpsWorks Stacks secara otomatis menyebarkan aplikasi dari jenis aplikasi standar ke lapisan server aplikasi bawaan yang sesuai. Untuk menerapkan aplikasi ke lapisan kustom, Anda harus menerapkan resep Deploy kustom yang mengunduh file aplikasi dari repositori ke lokasi yang sesuai pada instance. Namun, Anda sering dapat membatasi jumlah kode yang harus Anda tulis dengan

menggunakan buku masak deploy bawaan untuk menangani beberapa aspek penerapan. Misalnya, jika Anda menyimpan file Anda di salah satu repositori yang didukung, buku masak bawaan dapat menangani detail pengunduhan file dari repositori ke instance lapisan.

`tomcat::deploy`Resep ini dimaksudkan untuk ditetapkan ke peristiwa siklus hidup 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
...
```

`tomcat::deploy`Resepnya menggunakan buku masak penerapan bawaan untuk aspek penerapan yang tidak spesifik untuk aplikasi. `deploy`Resep (yang merupakan singkatan dari `deploy::default` resep bawaan) adalah resep bawaan yang menangani detail pengaturan pengguna, grup, dan sebagainya, berdasarkan data dari atribut. `deploy`

Resepnya menggunakan dua definisi Chef bawaan, `opsworks_deploy_dir` dan `opworks_deploy` untuk menginstal aplikasi.

`opsworks_deploy_dir`Definisi mengatur struktur direktori, berdasarkan data dari JSON penerapan aplikasi. Definisi pada dasarnya adalah cara yang nyaman untuk mengemas definisi sumber daya, dan terletak di `definitions` direktori buku masak. Resep dapat menggunakan definisi seperti sumber daya, tetapi definisi itu sendiri tidak memiliki penyedia terkait, hanya sumber daya yang termasuk dalam definisi. Anda dapat menentukan variabel dalam resep, yang diteruskan ke definisi sumber daya yang mendasarinya. Set `tomcat::deploy` resep`user`,`group`, dan `path` variabel berdasarkan data dari penyebaran JSON. Mereka diteruskan ke sumber daya direktori definisi, yang mengelola direktori.

> **Note**
>
> Pengguna dan grup aplikasi yang Anda gunakan ditentukan oleh `[:opsworks][:deploy_user][:group]` atribut `[:opsworks][:deploy_user][:user]` dan, yang ditentukan dalam file atribut buku masak penerapan bawaan. `deploy.rb` Nilai default `[:opsworks][:deploy_user][:user]` adalah `deploy`. Nilai default `[:opsworks][:deploy_user][:group]` tergantung pada sistem operasi instans:
>
> - Untuk instance Ubuntu, grup defaultnya adalah`www-data`.
> - Untuk instance Amazon Linux yang merupakan anggota lapisan Rails App Server yang menggunakan Nginx dan Unicorn, grup defaultnya adalah. `nginx`
> - Untuk semua instance Amazon Linux lainnya, grup defaultnya adalah`apache`.
>
> Anda dapat mengubah pengaturan dengan menggunakan JSON kustom atau file atribut kustom untuk mengganti atribut yang sesuai. Untuk informasi selengkapnya, lihat Mengesampingkan Atribut.

Definisi lain,`opsworks_deploy`, menangani detail memeriksa kode aplikasi dan file terkait dari repositori dan menerapkannya ke instance, berdasarkan data dari atribut. `deploy` Anda dapat menggunakan definisi ini untuk semua jenis aplikasi; detail penerapan seperti nama direktori ditentukan di konsol atau melalui API dan dimasukkan ke dalam `deploy` atribut. Namun, hanya `opsworks_deploy` berfungsi untuk empat jenis repositori yang didukung: Git, Subversion, S3, dan HTTP. Anda harus menerapkan kode ini sendiri jika Anda ingin menggunakan jenis repositori yang berbeda.

Anda menginstal file aplikasi di `webapps` direktori Tomcat. Praktik yang umum adalah menyalin file secara langsung ke`webapps`. Namun, penerapan AWS OpsWorks Stacks dirancang untuk mempertahankan hingga lima versi aplikasi pada sebuah instans, sehingga Anda dapat memutar kembali ke versi sebelumnya jika perlu. AWS OpsWorks Oleh karena itu, tumpukan melakukan hal berikut:

1. Menyebarkan aplikasi ke direktori berbeda yang namanya berisi cap waktu, seperti. `/srv/www/my_1st_jsp/releases/20130731141527`
2. Membuat symlink bernama`current`, seperti`/srv/www/my_1st_jsp/current`, ke direktori unik ini.

3. Jika belum ada, buat symlink dari `webapps` direktori ke `current` symlink yang dibuat di Langkah 2.

Jika Anda perlu memutar kembali ke versi sebelumnya, ubah `current` symlink untuk menunjuk ke direktori berbeda yang berisi stempel waktu yang sesuai, misalnya, dengan mengubah target tautan. `/srv/www/my_1st_jsp/current`

Bagian tengah `tomcat::deploy` mengatur symlink.

```
 ...
 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
 ...
```

Resep pertama menciptakan dua variabel, `current_dir` dan `webapp_dir` untuk mewakili `current` dan `webapp` direktori, masing-masing. Kemudian menggunakan `link` sumber daya untuk `webapp_dir` ditautkan`current_dir`. `deploy::default`Resep AWS OpsWorks Stacks membuat beberapa direktori rintisan yang tidak diperlukan untuk contoh ini, sehingga bagian tengah kutipan menghapusnya.

Bagian terakhir dari `tomcat::deploy` restart layanan Tomcat, jika perlu.

```
 ...
 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'
```

Resep pertama berjalan`tomcat::service`, untuk memastikan bahwa layanan ditentukan untuk menjalankan Chef ini. Kemudian menggunakan sumber daya eksekusi untuk memberi tahu layanan untuk memulai ulang, tetapi hanya jika `['tomcat']['auto_deploy']` diatur ke`'true'`. Jika tidak, Tomcat mendengarkan perubahan dalam `webapps` direktorinya, yang membuat restart layanan Tomcat eksplisit tidak perlu.

> **Note**
>
> `execute`Sumber daya tidak benar-benar mengeksekusi sesuatu yang substantif; `/bin/true` adalah skrip shell dummy yang hanya mengembalikan kode sukses. Ini digunakan di sini hanya sebagai cara mudah untuk menghasilkan pemberitahuan restart. Seperti disebutkan sebelumnya, menggunakan notifikasi memastikan bahwa layanan tidak dimulai ulang terlalu sering.

Terakhir, `tomcat::deploy` run`tomcat::context`, yang memperbarui file konfigurasi konteks aplikasi web jika Anda telah mengubah database back end.

Buat Stack dan Jalankan Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menunjukkan bagaimana menggunakan buku masak Tomcat untuk mengimplementasikan pengaturan tumpukan dasar yang menjalankan aplikasi halaman server Java sederhana (JSP) bernama SimpleJSP. Tumpukan terdiri dari lapisan kustom berbasis TomCustom TomCat bernama dan lapisan MySQL. SimpleJSP dikerahkan ke TomCustom dan menampilkan beberapa informasi dari database MySQL. Jika Anda belum terbiasa dengan dasar-dasar cara menggunakan AWS OpsWorks Stacks, Anda harus membaca Memulai dengan Chef 11 Linux Stacks terlebih dahulu.

Aplikasi SimpleJSP

Aplikasi SimpleJSP menunjukkan dasar-dasar cara mengatur koneksi database dan mengambil data dari database MySQL stack.

```
<html>
  <head>
    <title>DB Access</title>
  </head>
  <body>
    <%@ page language="java" import="java.sql.*,javax.naming.*,javax.sql.*" %>
    <%
      StringBuffer output = new StringBuffer();
      DataSource ds = null;
      Connection con = null;
      Statement stmt = null;
      ResultSet rs = null;
      try {
        Context initCtx = new InitialContext();
        ds = (DataSource) initCtx.lookup("java:comp/env/jdbc/mydb");
        con = ds.getConnection();
        output.append("Databases found:<br>");
        stmt = con.createStatement();
        rs = stmt.executeQuery("show databases");
        while (rs.next()) {
          output.append(rs.getString(1));
          output.append("<br>");
        }
      }
      catch (Exception e) {
        output.append("Exception: ");
        output.append(e.getMessage());
        output.append("<br>");
      }
      finally {
```

```
        try {
          if (rs != null) {
            rs.close();
          }
          if (stmt != null) {
            stmt.close();
          }
          if (con != null) {
            con.close();
          }
        }
        catch (Exception e) {
          output.append("Exception (during close of connection): ");
          output.append(e.getMessage());
          output.append("<br>");
        }
      }
    %>
    <%= output.toString() %>
  </body>
</html>
```

SimpleJSP menggunakan `DataSource` objek untuk berkomunikasi dengan database MySQL. Tomcat menggunakan data dalam file konfigurasi konteks aplikasi web untuk membuat dan menginisialisasi `DataSource` objek dan mengikatnya ke nama logis. Kemudian mendaftarkan nama logis dengan layanan penamaan Java Naming and Directory Interface (JNDI). Untuk mendapatkan instance dari `DataSource` objek yang sesuai, Anda membuat `InitialContext` objek dan meneruskan nama logis sumber daya ke `lookup` metode objek, yang mengambil objek yang sesuai. Nama logis contoh SimpleJSP,`java:comp/env/jdbc/mydb`, memiliki komponen berikut:

- Namespace root,`java`, yang dipisahkan dari sisa nama dengan titik dua (:).
- Setiap ruang nama tambahan, dipisahkan oleh garis miring maju (/).

  Tomcat secara otomatis menambahkan sumber daya ke `comp/env` namespace.
- Nama resource, yang didefinisikan dalam file konfigurasi konteks aplikasi web dan dipisahkan dari ruang nama dengan garis miring ke depan.

  Nama sumber daya untuk contoh ini adalah`jdbc/mydb`.

Untuk membuat koneksi ke database, SimpleJSP melakukan hal berikut:

1. Memanggil `getConnection` metode `DataSource` objek, yang mengembalikan `Connection` objek.
2. Memanggil `createStatement` metode `Connection` objek untuk membuat `Statement` objek, yang Anda gunakan untuk berkomunikasi dengan database.
3. Berkomunikasi dengan database dengan memanggil `Statement` metode yang sesuai.

   SimpleJSP memanggil `executeQuery` untuk mengeksekusi query SHOW DATABASES, yang mencantumkan database server.

`executeQuery`Metode mengembalikan `ResultSet` objek, yang berisi hasil query. SimpleJSP mendapatkan nama database dari `ResultSet` objek yang dikembalikan dan menggabungkan mereka untuk membuat string output. Akhirnya, contoh menutup`ResultSet`,`Statement`, dan `Connection` objek. Untuk informasi lebih lanjut tentang JSP dan JDBC, lihat JavaServer Pages Technology dan JDBC Basics, masing-masing.

Untuk menggunakan SimpleJSP dengan tumpukan, Anda harus memasukkannya ke dalam repositori. Anda dapat menggunakan salah satu repositori yang didukung, tetapi untuk menggunakan SimpleJSP dengan tumpukan contoh yang dibahas di bagian berikut, Anda harus memasukkannya ke dalam arsip S3 publik. Untuk informasi tentang cara menggunakan repositori standar lainnya, lihat. Repositori

Untuk menempatkan SimpleJSP di repositori arsip S3

1. Salin kode contoh ke file bernama `simplejsp.jsp` dan letakkan file dalam direktori bernama`simplejsp`.
2. Buat `.zip` arsip `simplejsp` direktori.
3. Buat bucket Amazon S3 publik, unggah `simplejsp.zip` ke bucket, dan buat file menjadi publik.

   Untuk penjelasan tentang cara melakukan tugas ini, lihat Memulai Dengan Amazon Simple Storage Service.

Buat Stack

Untuk menjalankan SimpleJSP Anda memerlukan tumpukan dengan lapisan berikut.

- Lapisan MySQL, yang mendukung server MySQL back end.
- Lapisan khusus yang menggunakan buku masak Tomcat untuk mendukung instance server Tomcat.

Untuk membuat tumpukan

1. Di dasbor AWS OpsWorks Stacks, klik Add Stack untuk membuat tumpukan baru dan klik Advanced >> untuk menampilkan semua opsi. Konfigurasikan tumpukan sebagai berikut.

   - Nama —Nama tumpukan yang ditentukan pengguna; contoh ini menggunakan. TomStack
   - Gunakan buku masak Chef kustom —Atur sakelar ke Ya, yang menampilkan beberapa opsi tambahan.
   - Jenis repositori —Git.
   - URL Repositori —. `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`
   - Custom Chef JSON —Tambahkan JSON berikut:

     ```
     {
       "tomcat": {
         "base_version": 7,
         "java_opts": "-Djava.awt.headless=true -Xmx256m"
       },
       "datasources": {
         "ROOT": "jdbc/mydb"
       }
     }
     ```

   Untuk opsi yang tersisa, Anda dapat menerima default.

   JSON kustom melakukan hal berikut:

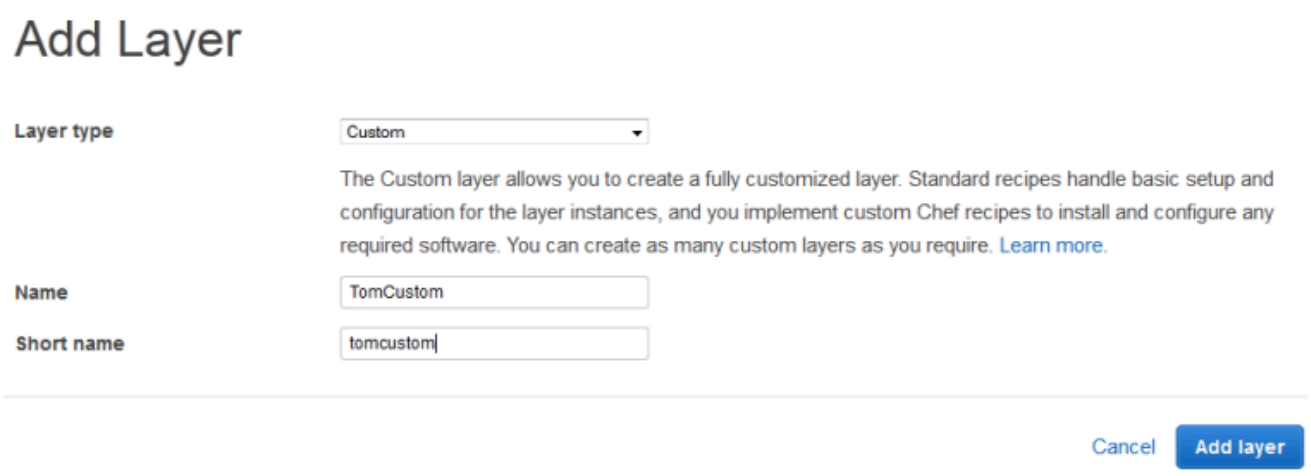
   - Mengganti `['base_version']` atribut buku masak Tomcat untuk mengatur versi Tomcat ke 7; nilai defaultnya adalah 6.
   - Mengganti `['java_opts']` atribut buku masak Tomcat untuk menentukan bahwa instance tanpa kepala dan menyetel ukuran heap maksimum JVM ke 256MB; nilai default tidak menetapkan opsi untuk instance yang menjalankan Amazon Linux.

- Menentukan nilai `['datasources]` atribut, yang menetapkan nama sumber daya JDBC (jdbc/mydb) ke nama konteks aplikasi web (ROOT), seperti yang dibahas dalam. kucing jantan: :konteks

  Atribut terakhir ini tidak memiliki nilai default; Anda harus mengaturnya dengan JSON kustom.



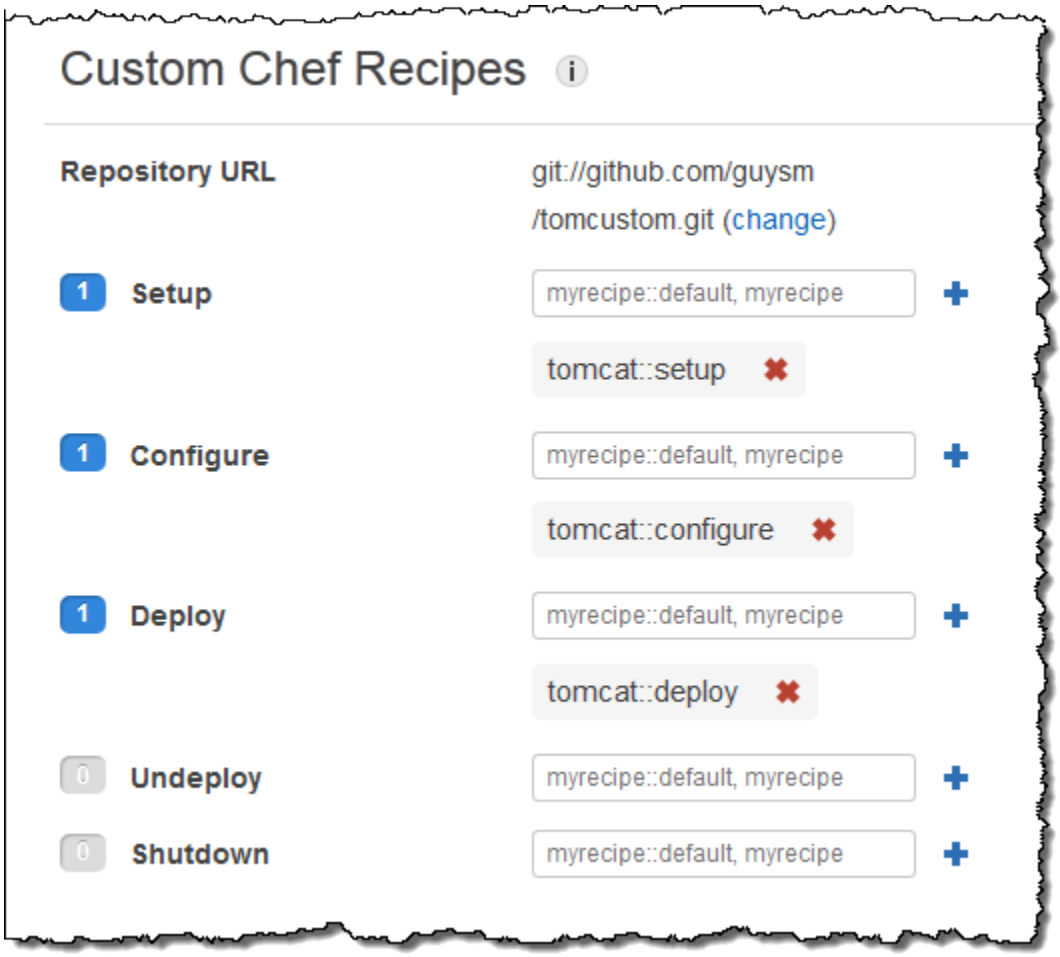
2. Klik Tambahkan layer. Untuk tipe Layer, pilih MySQL. Kemudian klik Add Layer.
3. Klik Instans di panel navigasi dan kemudian klik Tambahkan instance. Klik Add Instance untuk menerima default. Pada baris untuk contoh, klik mulai.
4. Kembali ke halaman Layers dan klik + Layer untuk menambahkan layer. Untuk jenis Layer, klik Custom .s Contoh menggunakan **`TomCustom`** dan **`tomcustom`** sebagai nama layer dan nama pendek, masing-masing.



5. Pada halaman Layers, untuk layer kustom, klik Resep dan kemudian klik Edit. Di bawah Resep Koki Kustom, tetapkan resep buku masak Tomcat ke acara siklus hidup lapisan, sebagai berikut:

   - Untuk Pengaturan, ketik **`tomcat::setup`** dan klik +.
   - Untuk Konfigurasi, ketik **`tomcat::configure`** dan klik +.
   - Untuk Deploy, ketik **`tomcat::deploy`** dan klik +. Kemudian klik Simpan.

   .



6. Klik Aplikasi di panel navigasi dan kemudian klik Tambahkan aplikasi. Tentukan opsi berikut, lalu klik Tambahkan Aplikasi:

   - Nama —Nama aplikasi; contohnya menggunakan SimpleJSP dan nama pendek yang dihasilkan oleh AWS OpsWorks Stacks adalah simplejsp.
   - Jenis aplikasi —Atur opsi ini ke Lainnya.

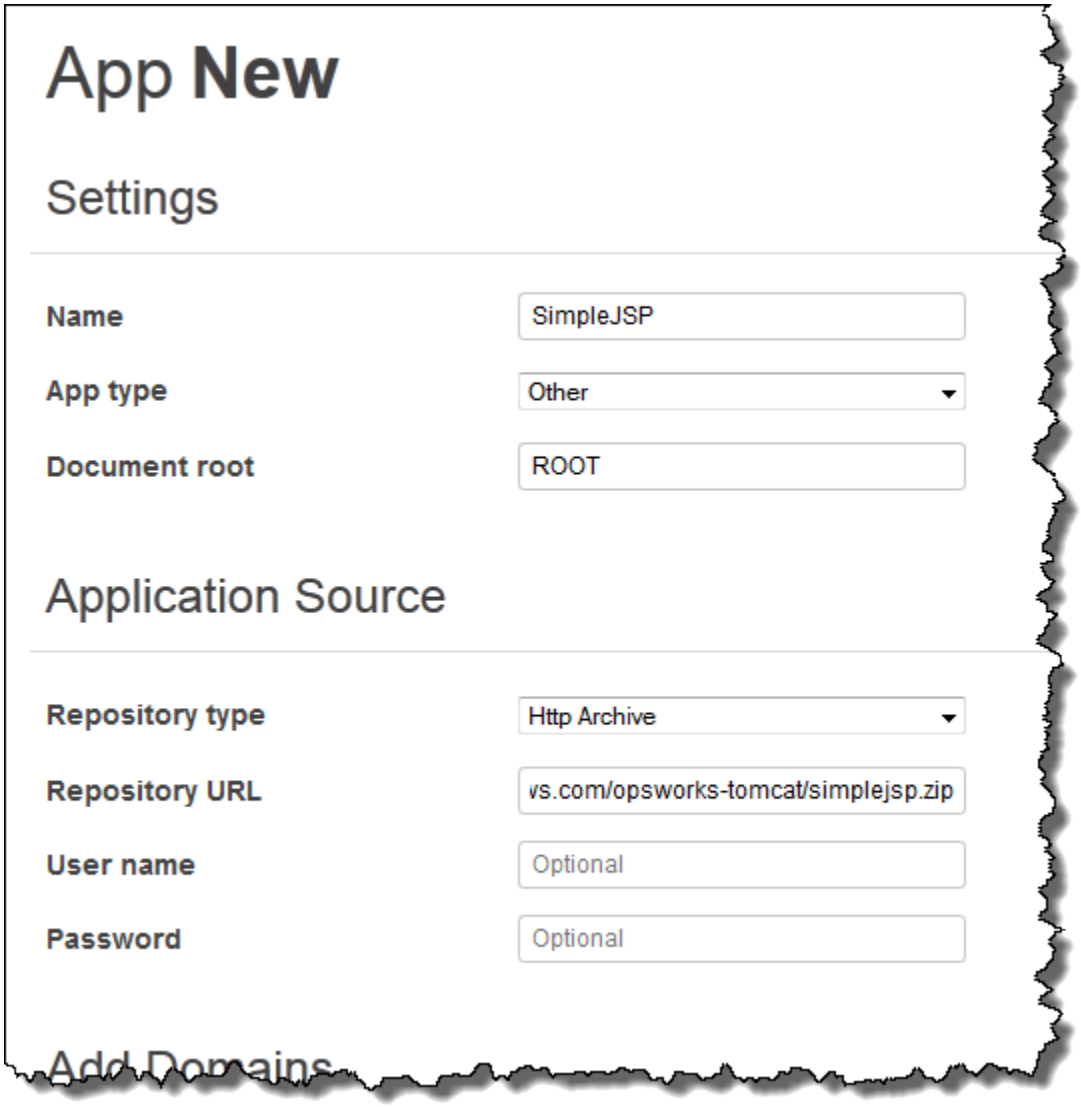
AWS OpsWorks Stacks secara otomatis menerapkan tipe aplikasi standar ke instance server terkait. Jika Anda menyetel jenis Aplikasi ke yang lain, AWS OpsWorks Stacks cukup menjalankan resep Deploy, dan memungkinkan mereka menangani penerapan.

- Root dokumen —Atur opsi ini ke**ROOT**.

  Nilai root Document menentukan nama konteks.

- Jenis repositori —Atur opsi ini ke Arsip S3.
- URL Repositori —Setel ini ke URL Amazon S3 aplikasi yang Anda buat sebelumnya.

Gunakan pengaturan default untuk opsi lain.



7. Gunakan halaman Instances untuk menambahkan instance ke TomCustom layer dan memulainya. AWS OpsWorks Stacks secara otomatis menjalankan resep Deploy pada instance baru setelah resep Setup selesai, jadi memulai instance juga menyebarkan SimpleJSP.
8. Saat TomCustom instans sedang online, klik nama instance pada halaman Instances untuk melihat detailnya. Salin alamat IP publik. Kemudian buat URL sebagai berikut: http:///tc/*publicIP*. *appname.jsp* Sebagai contoh, URL ini akan terlihat seperti**http://50.218.191.172/tc/simplejsp.jsp**.

   > **Note**
   >
   > URL Apache yang meneruskan permintaan ke Tomcat diatur ke atribut default['tomcat']['apache_tomcat_bind_path'],. /tc/ Root dokumen SimpleJSP diatur ke R00T mana merupakan nilai khusus yang diselesaikan. / Oleh karena itu URL adalah "... /tc/simplejsp.jsp".

9. Tempel URL dari langkah sebelumnya ke browser Anda. Anda akan melihat yang berikut ini:

```
Databases found:
information_schema
simplejsp
test
```

> **Note**
>
> Jika tumpukan Anda memiliki instance MySQL AWS OpsWorks , Stacks secara otomatis membuat database untuk setiap aplikasi, dinamai dengan nama pendek aplikasi.

Konfigurasi Stack dan Atribut Deployment

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Ketika AWS OpsWorks Stacks menjalankan perintah pada instance—misalnya, perintah deploy sebagai respons terhadap peristiwa siklus hidup Deploy—ia menambahkan satu set atribut ke objek node instance yang menjelaskan konfigurasi tumpukan saat ini. Untuk peristiwa Deploy dan perintah stack Execute Recipes, AWS OpsWorks Stacks menginstal atribut deploy, yang menyediakan beberapa informasi penerapan tambahan. Untuk informasi lebih lanjut tentang objek node, lihatMengesampingkan Atribut. Untuk daftar konfigurasi tumpukan dan atribut penerapan yang umum digunakan, termasuk nama node yang sepenuhnya memenuhi syarat, lihat Konfigurasi Stack dan Atribut Deployment: Linux danAtribut Buku Masak Bawaan.

Note

Pada tumpukan Linux, Anda bisa mendapatkan daftar lengkap atribut ini, diformat sebagai objek JSON, dengan menggunakan perintah get_json agen CLI.

Bagian berikut menunjukkan atribut yang terkait dengan peristiwa Configure dan event Deploy untuk tumpukan sederhana, yang terdiri dari berikut ini:

- Lapisan PHP App Server dengan dua instance
- HAProxy Lapisan dengan satu contoh

Contohnya adalah dari salah satu instance PHP App Server, php-app1. Untuk kenyamanan, atribut diformat sebagai objek JSON. Struktur objek memetakan ke nama atribut yang sepenuhnya memenuhi syarat. Misalnya, `node[:opsworks][:ruby_version]` atribut muncul sebagai berikut dalam representasi JSON.

```
{
  "opsworks": {
    ...
    "ruby_version": "1.8.7",
    ...
  }
}
```

Topik

- Konfigurasikan Atribut
- Atribut Penerapan

Konfigurasikan Atribut

Objek JSON berikut menunjukkan atribut untuk acara Configure, yang terjadi pada setiap instance dalam tumpukan ketika sebuah instance online atau offline. Atribut termasuk atribut konfigurasi tumpukan bawaan dan atribut JSON khusus apa pun yang ditentukan untuk tumpukan sebelum acara (tidak ada dalam contoh ini). Ini telah diedit untuk waktu yang lama. Untuk penjelasan rinci tentang berbagai atribut, lihat Konfigurasi Stack dan Atribut Deployment: Linux danAtribut Buku Masak Bawaan.

```
{
  "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": {
  }
}
```

Sebagian besar informasi berada di bawah `opsworks` atribut, yang sering disebut sebagai namespace. Daftar berikut menjelaskan atribut kunci:

- `layers`atribut — Satu set atribut, yang masing-masing menggambarkan konfigurasi salah satu lapisan tumpukan.

  Lapisan diidentifikasi oleh nama pendek mereka, `php-app` dan `lb` untuk contoh ini. Untuk informasi selengkapnya tentang shortname untuk layer lain, lihatAWS OpsWorks Referensi Lapisan Tumpukan.
- `instances`atribut — Setiap lapisan memiliki `instances` elemen, yang mencakup atribut untuk setiap instance online lapisan, dinamai dengan nama pendek instance.

  Lapisan PHP App Server memiliki dua instance, `php-app1` dan`php-app2`. HAProxy Lapisan memiliki satu contoh,`lb1`.

  > **Note**
  >
  > `instances`Elemen hanya berisi instance yang berada dalam status online saat atribut stack dan deployment tertentu dibuat.

- Atribut instans - Setiap atribut instance berisi sekumpulan atribut yang menjadi ciri instance, seperti alamat IP pribadi instans dan nama DNS pribadi. Untuk singkatnya, contoh hanya menampilkan `php-app2` atribut secara rinci; yang lain berisi informasi serupa.
- `applications`— Daftar aplikasi yang digunakan, tidak digunakan dalam contoh ini.
- `stack`— Nama tumpukan; `MyStack` dalam contoh ini.
- `instance`— Contoh bahwa atribut ini diinstal pada; `php-app1` dalam contoh ini. Resep dapat menggunakan atribut ini untuk mendapatkan informasi tentang instance yang mereka jalankan, seperti alamat IP publik instans.
- `activity`— Aktivitas yang menghasilkan atribut; sebuah peristiwa Configure dalam contoh ini.
- `rails_stack`— Tumpukan Rails untuk tumpukan yang menyertakan lapisan Rails App Server.
- `deployment`— Apakah atribut ini terkait dengan penerapan. Hal ini diatur untuk `null` untuk contoh ini karena mereka terkait dengan peristiwa Configure.
- `valid_client_activities`— Daftar aktivitas klien yang valid.

`opsworks`Atribut diikuti oleh beberapa atribut tingkat atas lainnya, termasuk yang berikut:

- `opsworks_custom_cookbooks`— Apakah buku masak khusus diaktifkan. Jika demikian, atribut menyertakan daftar resep khusus.
- `recipes`— Resep yang dijalankan oleh kegiatan ini.

- `opsworks_rubygems`— RubyGems Versi instans.
- `ssh_users`— Daftar pengguna SSH; tidak ada dalam contoh ini.
- `opsworks_bundler`— Versi bundler dan apakah itu diaktifkan.
- `deploy`— Informasi tentang aktivitas penyebaran; tidak ada dalam contoh ini.

Atribut Penerapan

Atribut untuk peristiwa Deploy atau perintah stack Execute Recipes terdiri dari konfigurasi tumpukan bawaan dan atribut penerapan, dan atribut tumpukan atau penerapan khusus apa pun (tidak ada untuk contoh ini). Objek JSON berikut menunjukkan atribut dari php-app1 yang terkait dengan peristiwa Deploy yang menerapkan aplikasi SimplePHP ke instance PHP stack. Sebagian besar objek terdiri dari atribut konfigurasi tumpukan yang mirip dengan yang untuk peristiwa Konfigurasi yang dijelaskan di bagian sebelumnya, jadi contoh berfokus terutama pada atribut khusus penerapan. Untuk penjelasan rinci tentang berbagai atribut, lihat Konfigurasi Stack dan Atribut Deployment: Linux danAtribut Buku Masak Bawaan.

```
{
   ...
  "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
      }
    }
```

```
  },
}
```

`opsworks`Atribut ini sebagian besar identik dengan contoh di bagian sebelumnya. Bagian berikut ini paling relevan dengan penerapan:

- `activity`— Peristiwa yang terkait dengan atribut ini; peristiwa Deploy dalam contoh ini.
- `applications`— Berisi sekumpulan atribut untuk setiap aplikasi yang menyediakan nama, nama slug, dan jenis aplikasi.

  Nama slug adalah nama pendek aplikasi, yang dihasilkan AWS OpsWorks Stacks dari nama aplikasi. Nama slug untuk SimplePHP adalah simplephp.
- `deployment`— ID penyebaran, yang secara unik mengidentifikasi penyebaran.

`deploy`Atribut mencakup informasi tentang aplikasi yang sedang digunakan. Misalnya, resep Deploy bawaan menggunakan data dalam `deploy` atribut untuk menginstal file di direktori yang sesuai dan membuat file koneksi database. `deploy`Atribut menyertakan satu atribut untuk setiap aplikasi yang di-deploy, dinamai dengan nama pendek aplikasi. Setiap atribut aplikasi menyertakan atribut berikut:

- `environment_variables`— Berisi variabel lingkungan apa pun yang telah Anda tentukan untuk aplikasi. Untuk informasi selengkapnya, lihat Variabel lingkungan.
- `domains`— Secara default, domain adalah nama pendek aplikasi, yang merupakan simplephpapp untuk contoh ini. Jika Anda telah menetapkan domain khusus, mereka muncul di sini juga. Untuk informasi selengkapnya, lihat Menggunakan Domain Kustom.
- `application`— Nama pendek aplikasi.
- `scm`— Elemen ini berisi informasi yang diperlukan untuk mengunduh file aplikasi dari repositorinya; sebuah repositori Git dalam contoh ini.
- `database`— Informasi database, jika tumpukan menyertakan lapisan database.
- `document_root`— Root dokumen, yang diatur ke `null` dalam contoh ini, menunjukkan bahwa root bersifat publik.
- `ssl_certificate_ca`,`ssl_support`, `ssl_certificate_key` — Menunjukkan apakah aplikasi memiliki dukungan SSL. Jika demikian, `ssl_certificate_ca` atribut `ssl_certificate_key` dan diatur ke sertifikat yang sesuai.
- `deploy_to`— Direktori root aplikasi.

Buku masak 101

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan AWS OpsWorks Stacks tingkat produksi biasanya memerlukan beberapa penyesuaian, yang sering berarti menerapkan buku masak Chef khusus dengan satu atau lebih resep, file atribut, atau file templat. Topik ini adalah pengantar tutorial untuk mengimplementasikan buku masak untuk AWS OpsWorks Stacks.

Untuk informasi lebih lanjut tentang bagaimana AWS OpsWorks Stacks menggunakan buku masak, yang mencakup pengantar umum singkat untuk buku masak, lihat. Buku Masak dan Resep Untuk informasi tambahan tentang cara menerapkan dan menguji resep Chef, lihat Test-Driven Infrastructure with Chef, 2nd Edition.

Contoh tutorial dibagi menjadi dua bagian:

- Dasar-dasar Buku Masakadalah serangkaian contoh penelusuran yang ditujukan untuk pengguna yang tidak terbiasa dengan Chef; pengguna Chef berpengalaman dapat melewati bagian ini.

  Contoh memandu Anda melalui dasar-dasar bagaimana menerapkan buku masak untuk melakukan tugas-tugas umum, seperti menginstal paket atau membuat direktori. Untuk menyederhanakan proses, Anda akan menggunakan sepasang alat yang berguna, Vagrant dan Test Kitchen, untuk menjalankan sebagian besar contoh secara lokal di mesin virtual. Sebelum memulaiDasar-dasar Buku Masak, Anda harus membaca terlebih dahulu Vagrant dan Test Kitchen untuk mempelajari cara menginstal dan menggunakan alat-alat ini. Karena Test Kitchen belum mendukung Windows, contohnya semuanya untuk Linux, dengan catatan yang menunjukkan cara menyesuaikannya untuk Windows.
- Menerapkan Buku Masak untuk Tumpukan AWS OpsWorksmenjelaskan cara menerapkan resep untuk AWS OpsWorks Tumpukan, termasuk untuk tumpukan Windows.

  Ini juga mencakup beberapa s yang lebih maju seperti cara menggunakan Berkshelf untuk mengelola buku masak eksternal. Contohnya ditulis untuk pengguna Chef baru, seperti contoh

diDasar-dasar Buku Masak. Namun AWS OpsWorks Stacks bekerja sedikit berbeda dari server Chef, jadi kami menyarankan agar pengguna Chef berpengalaman setidaknya membaca bagian ini.

Vagrant dan Test Kitchen

Jika Anda bekerja dengan resep untuk instance Linux, Vagrant dan Test Kitchen adalah alat yang sangat berguna untuk pembelajaran dan pengembangan awal dan pengujian. Ini memberikan deskripsi singkat tentang Vagrant dan Test Kitchen, dan mengarahkan Anda ke petunjuk pemasangan dan panduan yang akan membuat Anda mengatur dan membiasakan Anda dengan dasar-dasar cara menggunakan alat. Meskipun Vagrant mendukung Windows, Test Kitchen tidak, jadi hanya contoh Linux yang disediakan untuk alat ini.

Gelandangan

Vagrant menyediakan lingkungan yang konsisten untuk mengeksekusi dan menguji kode pada mesin virtual. Ini mendukung berbagai lingkungan — disebut kotak Vagrant — yang masing-masing mewakili sistem operasi yang dikonfigurasi. Untuk AWS OpsWorks Stacks, lingkungan yang menarik didasarkan pada distribusi Ubuntu, Amazon, atau Red Hat Enterprise Linux (RHEL), jadi contohnya terutama menggunakan kotak Vagrant bernama. `opscode-ubuntu-12.04`

Vagrant tersedia untuk sistem Linux, Windows, dan Macintosh, sehingga Anda dapat menggunakan workstation pilihan Anda untuk mengimplementasikan dan menguji resep pada sistem operasi yang didukung. Contoh untuk bagian ini dibuat pada sistem Linux Ubuntu, tetapi menerjemahkan prosedur ke sistem Windows atau Macintosh sangat mudah.

Vagrant pada dasarnya adalah pembungkus untuk penyedia virtualisasi. Sebagian besar contoh menggunakan VirtualBoxpenyedia. VirtualBox gratis dan tersedia untuk sistem Linux, Windows, dan Macintosh. Panduan Vagrant memberikan instruksi instalasi jika Anda belum memilikinya VirtualBox di sistem Anda. Perhatikan bahwa Anda dapat menjalankan lingkungan berbasis Ubuntu VirtualBox, tetapi Amazon Linux hanya tersedia untuk instans Amazon EC2 . Namun, Anda dapat menjalankan sistem operasi serupa seperti CentOS on VirtualBox, yang berguna untuk pengembangan dan pengujian awal.

Untuk informasi tentang penyedia lain, lihat dokumentasi Vagrant. Secara khusus, penyedia `vagrant-aws` plug-in memungkinkan Anda untuk menggunakan Vagrant dengan instans Amazon EC2 . Penyedia ini sangat berguna untuk menguji resep di Amazon Linux, yang hanya tersedia di

EC2 instans Amazon. `vagrant-aws`Penyedia gratis, tetapi Anda harus memiliki akun AWS dan membayar sumber daya AWS apa pun yang Anda gunakan.

Pada titik ini, Anda harus melalui panduan Memulai Vagrant, yang menjelaskan cara menginstal Vagrant di workstation Anda dan mengajarkan Anda dasar-dasar cara menggunakan Vagrant. Perhatikan bahwa contoh dalam Bab ini tidak menggunakan repositori Git, sehingga Anda dapat menghilangkan bagian dari panduan tersebut jika Anda mau.

Uji Dapur

Test Kitchen menyederhanakan proses mengeksekusi dan menguji buku masak Anda di Vagrant. Sebagai masalah praktis, Anda jarang jika pernah perlu menggunakan Vagrant secara langsung. Test Kitchen melakukan tugas yang paling umum, termasuk:

- Meluncurkan sebuah instance di Vagrant.
- Mentransfer buku masak ke instance.
- Menjalankan resep buku masak pada instance.
- Menguji resep buku masak pada instance.
- Menggunakan SSH untuk masuk ke instance.

Alih-alih memasang permata Test Kitchen secara langsung, kami sarankan menginstal Chef DK. Selain Chef sendiri, paket ini termasuk Test Kitchen, Berkshelf ChefSpec, dan beberapa alat berguna lainnya.

Pada titik ini, Anda harus melalui panduan Memulai Test Kitchen, yang mengajarkan Anda dasar-dasar cara menggunakan Test Kitchen untuk mengeksekusi dan menguji resep.

> **Note**
>
> Contoh dalam Bab ini menggunakan Test Kitchen sebagai cara mudah untuk menjalankan resep. Jika mau, Anda dapat menghentikan panduan Memulai setelah menyelesaikan bagian Verifikasi Secara Manual, yang mencakup semua yang perlu Anda ketahui untuk contoh. Namun, Test Kitchen terutama merupakan platform pengujian yang mendukung kerangka pengujian seperti bash automated test system (BATS). Anda harus menyelesaikan sisa panduan di beberapa titik untuk mempelajari cara menggunakan Test Kitchen untuk menguji resep Anda.

Dasar-dasar Buku Masak

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menggunakan buku masak untuk menyelesaikan berbagai tugas. Topik berikut mengasumsikan bahwa Anda baru mengenal Chef, dan menjelaskan cara menggunakan buku masak untuk menyelesaikan beberapa tugas umum. Karena Test Kitchen belum mendukung Windows, contohnya semuanya untuk Linux, dengan catatan yang menunjukkan cara menyesuaikannya untuk Windows. Jika Anda baru mengenal Chef, kami sarankan untuk melihat contoh-contoh ini, bahkan jika Anda akan bekerja dengan Windows. Sebagian besar contoh dalam topik ini dapat digunakan pada instance Windows dengan beberapa perubahan sederhana, yang dicatat dalam contoh. Semua contoh berjalan di mesin virtual, jadi Anda bahkan tidak perlu memiliki komputer Linux. Cukup instal Vagrant dan Test Kitchen di workstation reguler Anda.

> **Note**
>
> Jika Anda ingin menjalankan resep ini pada instance Windows, pendekatan paling sederhana adalah membuat tumpukan Windows dan menjalankan resep di salah satu instance tumpukan. Untuk informasi selengkapnya tentang cara menjalankan resep pada instance AWS OpsWorks Stacks Windows, lihatMenjalankan Resep pada Instance Windows.

Sebelum melanjutkan, pastikan Anda telah menginstal Vagrant dan Test Kitchen, dan melalui panduan Memulai mereka. Untuk informasi selengkapnya, lihat Vagrant dan Test Kitchen.

Topik

- Struktur Resep
- Contoh 1: Menginstal Paket
- Contoh 2: Mengelola Pengguna
- Contoh 3: Membuat Direktori

- Contoh 4: Menambahkan Kontrol Aliran
- Contoh 5: Menggunakan Atribut
- Contoh 6: Membuat File
- Contoh 7: Menjalankan Perintah dan Skrip
- Contoh 8: Mengelola Layanan
- Contoh 9: Menggunakan EC2 Instans Amazon
- Langkah Berikutnya

Struktur Resep

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Buku masak pada dasarnya adalah seperangkat resep, yang dapat melakukan berbagai tugas pada sebuah instance. Untuk memperjelas cara menerapkan resep, ada baiknya melihat contoh sederhana. Berikut ini adalah resep pengaturan untuk HAProxylapisan bawaan. Fokus saja pada struktur keseluruhan pada saat ini dan jangan terlalu khawatir tentang detailnya; mereka akan dibahas dalam contoh berikutnya.

```
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**
>
> Untuk ini dan contoh lain dari resep kerja dan file terkait, lihat resep bawaan AWS OpsWorks Tumpukan.

Contoh ini menyoroti elemen resep utama, yang dijelaskan di bagian berikut.

Topik

- Sumber daya
- Kontrol Aliran
- Termasuk Resep

Sumber daya

Resep sebagian besar terdiri dari satu set sumber daya Chef. Masing-masing menentukan aspek tertentu dari status akhir instance, seperti paket yang akan diinstal atau layanan yang akan dimulai. Contohnya memiliki empat sumber daya:

- `package`Sumber daya, yang mewakili paket yang diinstal, HAProxy server untuk contoh ini.
- `service`Sumber daya, yang mewakili layanan, HAProxy layanan untuk contoh ini.
- Dua `template` sumber daya, yang mewakili file yang akan dibuat dari template tertentu, dua file HAProxy konfigurasi untuk contoh ini.

Sumber daya menyediakan cara deklaratif untuk menentukan status instance. Di belakang layar, setiap sumber daya memiliki penyedia terkait yang melakukan tugas yang diperlukan, seperti menginstal paket, membuat dan mengonfigurasi direktori, memulai layanan, dan sebagainya. Jika detail tugas bergantung pada sistem operasi tertentu, sumber daya memiliki banyak penyedia dan menggunakan yang sesuai untuk sistem. Misalnya, pada sistem Red Hat Linux yang digunakan `package` penyedia `yum` untuk menginstal paket. Pada sistem Ubuntu Linux, `package` penyedia menggunakan`apt-get`.

Anda menerapkan sumber daya sebagai blok kode Ruby dengan format umum berikut.

```
resource_type "resource_name" do
  attribute1 'value1'
  attribute2 'value2'
  ...
  action :action_name
  notifies : action 'resource'
end
```

Elemen-elemennya adalah:

Jenis sumber daya

(Wajib) Contoh ini mencakup tiga jenis sumber daya`package`,`service`,, dan`template`.

Nama sumber daya

(Wajib) Nama mengidentifikasi sumber daya tertentu dan kadang-kadang digunakan sebagai nilai default untuk salah satu atribut. Dalam contoh, `package` merupakan sumber daya paket bernama `haproxy` dan `template` sumber daya pertama mewakili file konfigurasi bernama`/etc/default/haproxy`.

Atribut

(Opsional) Atribut menentukan konfigurasi sumber daya dan bervariasi tergantung pada jenis sumber daya dan bagaimana Anda ingin mengonfigurasi sumber daya.

- `template`Sumber daya contoh secara eksplisit mendefinisikan sekumpulan atribut yang menentukan sumber, pemilik, grup, dan mode file yang dibuat.
- Contoh `package` dan `service` sumber daya tidak secara eksplisit mendefinisikan atribut apa pun.

Nama sumber daya biasanya merupakan nilai default untuk atribut yang diperlukan dan terkadang semua yang diperlukan. Misalnya, nama sumber daya adalah nilai default untuk `package_name` atribut `package` sumber daya, yang merupakan satu-satunya atribut yang diperlukan.

Ada juga beberapa atribut khusus yang disebut atribut penjaga, yang menentukan kapan penyedia sumber daya akan mengambil tindakan. Misalnya, `only_if` atribut mengarahkan penyedia sumber daya untuk mengambil tindakan hanya jika kondisi tertentu terpenuhi. HAProxy Resepnya tidak menggunakan atribut penjaga, tetapi digunakan oleh beberapa contoh berikut.

Tindakan dan Pemberitahuan

(Opsional) Tindakan dan pemberitahuan menentukan tugas apa yang harus dilakukan penyedia.

- `action`mengarahkan penyedia untuk mengambil tindakan tertentu, seperti menginstal atau membuat.

  Setiap sumber daya memiliki serangkaian tindakan yang bergantung pada sumber daya tertentu, salah satunya adalah tindakan default. Dalam contoh, tindakan `package` sumber daya adalah`install`, yang mengarahkan penyedia untuk menginstal paket. `template`Sumber daya pertama tidak memiliki `action` elemen, sehingga penyedia mengambil `create` tindakan default.

- `notifies`mengarahkan penyedia sumber daya lain untuk melakukan tindakan, tetapi hanya jika status sumber daya telah berubah.

  `notifies`biasanya digunakan dengan sumber daya seperti `template` dan `file` untuk melakukan tugas-tugas seperti memulai kembali layanan setelah memodifikasi file konfigurasi. Sumber daya tidak memiliki pemberitahuan default. Jika Anda menginginkan notifikasi, sumber daya harus memiliki `notifies` elemen eksplisit. Dalam HAProxy resep, `template` sumber daya kedua memberi tahu sumber daya haproxy `service` untuk memulai ulang HAProxy layanan jika file konfigurasi terkait telah berubah.

Sumber daya terkadang bergantung pada sistem operasi.

- Beberapa sumber daya hanya dapat digunakan pada sistem Linux atau Windows.

  Misalnya, paket menginstal paket pada sistem Linux dan windows_package menginstal paket pada sistem Windows.

- Beberapa sumber daya dapat digunakan dengan sistem operasi apa pun, tetapi memiliki atribut yang spesifik untuk sistem tertentu.

  Misalnya, sumber daya file dapat digunakan pada sistem Linux atau Windows, tetapi memiliki set atribut terpisah untuk mengonfigurasi izin.

Untuk deskripsi sumber daya standar, termasuk atribut, tindakan, dan pemberitahuan yang tersedia untuk setiap sumber daya, lihat Tentang Sumber Daya dan Penyedia.

Kontrol Aliran

Karena resep adalah aplikasi Ruby, Anda dapat menggunakan struktur kontrol Ruby untuk memasukkan kontrol aliran ke dalam resep. Misalnya, Anda dapat menggunakan logika bersyarat Ruby agar resep berperilaku berbeda pada sistem yang berbeda. HAProxy Resep termasuk `if` blok yang menggunakan `template` sumber daya untuk membuat file konfigurasi, tetapi hanya jika resep berjalan pada sistem Debian atau Ubuntu.

Skenario umum lainnya adalah menggunakan loop untuk mengeksekusi sumber daya beberapa kali dengan pengaturan atribut yang berbeda. Misalnya, Anda dapat membuat satu set direktori dengan menggunakan loop untuk mengeksekusi `directory` sumber daya beberapa kali dengan nama direktori yang berbeda.

> **Note**
>
> Jika Anda tidak terbiasa dengan Ruby, lihat Just Enough Ruby for Chef, yang mencakup apa yang perlu Anda ketahui untuk sebagian besar resep.

Termasuk Resep

`include_recipe`termasuk resep lain dalam kode Anda, yang memungkinkan Anda untuk memodulasi resep Anda dan menggunakan kembali kode yang sama dalam beberapa resep. Saat Anda menjalankan resep host, Chef mengganti setiap `include_recipe` elemen dengan kode resep yang ditentukan sebelum mengeksekusi resep host. Anda mengidentifikasi resep yang disertakan dengan menggunakan `cookbook_name::recipe_name` sintaks Chef standar, di mana `recipe_name` menghilangkan ekstensi. `.rb` Contohnya termasuk satu resep,`haproxy::service`, yang mewakili HAProxy layanan.

> **Note**
>
> Jika Anda menggunakan `include_recipe` resep yang berjalan pada Chef 11.10 dan yang lebih baru untuk menyertakan resep dari buku masak lain, Anda harus menggunakan `depends` pernyataan untuk mendeklarasikan ketergantungan dalam file buku masak. `metadata.rb` Untuk informasi selengkapnya, lihat Resep Penerapan: Chef 11.10.

Contoh 1: Menginstal Paket

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Instalasi paket adalah salah satu penggunaan resep yang lebih umum dan bisa sangat sederhana, tergantung pada paketnya. Misalnya, resep berikut menginstal Git pada sistem Linux.

```
package 'git' do
  action :install
end
```

Sumber `package`daya menangani instalasi paket. Untuk contoh ini, Anda tidak perlu menentukan atribut apa pun. Nama sumber daya adalah nilai default untuk `package_name` atribut, yang mengidentifikasi paket. `install`Tindakan mengarahkan penyedia untuk menginstal paket. Anda dapat membuat kode lebih sederhana dengan melewatkan`install`; itu adalah tindakan default `package` sumber daya. Ketika Anda menjalankan resep, Chef menggunakan penyedia yang sesuai untuk menginstal paket. Pada sistem Ubuntu yang akan Anda gunakan sebagai contoh, penyedia menginstal Git dengan menelepon`apt-get`.

> **Note**
>
> Menginstal perangkat lunak pada sistem Windows memerlukan prosedur yang agak berbeda. Untuk informasi selengkapnya, lihat Menginstal Perangkat Lunak Windows.

Untuk menggunakan Test Kitchen untuk menjalankan resep ini di Vagrant, Anda harus terlebih dahulu menyiapkan buku masak dan menginisialisasi dan mengkonfigurasi Test Kitchen. Berikut ini adalah untuk sistem Linux, tetapi prosedurnya pada dasarnya serupa untuk sistem Windows dan Macintosh. Mulailah dengan membuka jendela Terminal; semua contoh dalam Bab ini menggunakan alat baris perintah.

Untuk menyiapkan buku masak

1. Di direktori home Anda, buat subdirektori bernama`opsworks_cookbooks`, yang akan berisi semua buku masak untuk chapter ini. Kemudian buat subdirektori untuk buku masak ini bernama `installpkg` dan arahkan ke sana.
2. Di`installpkg`, buat file bernama `metadata.rb` yang berisi kode berikut.

   ```
   name "installpkg"
   version "0.1.0"
   ```

   Untuk mempermudah, contoh dalam Bab ini hanya menentukan nama buku masak dan versi, tetapi `metadata.rb` dapat berisi berbagai metadata buku masak. Untuk informasi lebih lanjut, lihat Tentang Metadata Buku Masak.

   > **Note**
   >
   > Pastikan untuk membuat `metadata.rb` sebelum Anda menginisialisasi Test Kitchen; itu menggunakan data untuk membuat file konfigurasi default.

3. Di`installpkg`, jalankan`kitchen init`, yang menginisialisasi Test Kitchen dan menginstal driver Vagrant default.
4. `kitchen init`Perintah membuat file konfigurasi YAMB dalam `installpkg` nama`.kitchen.yml`. Buka file dalam editor teks favorit Anda. `.kitchen.yml`File tersebut menyertakan `platforms` bagian yang menentukan sistem mana yang akan menjalankan resep. Test Kitchen membuat instance dan menjalankan resep yang ditentukan pada setiap platform.

> **Note**
>
> Secara default, Test Kitchen menjalankan resep satu platform pada satu waktu. Jika Anda menambahkan `-p` argumen ke perintah apa pun yang membuat instance, Test Kitchen akan menjalankan resep di setiap platform, secara paralel.

Satu platform sudah cukup untuk contoh ini, jadi edit `.kitchen.yml` untuk menghapus `centos-6.4` platform. `.kitchen.yml`File Anda sekarang akan terlihat seperti ini:

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo

platforms:
  - name: ubuntu-12.04

suites:
  - name: default
    run_list:
      - recipe[installpkg::default]
    attributes:
```

Test Kitchen hanya menjalankan resep-resep yang ada dalam daftar `.kitchen.yml` run. Anda mengidentifikasi resep dengan menggunakan `[`*`cookbook_name`*`::`*`recipe_name`*`]` format, di mana *`recipe_name`* menghilangkan `.rb` ekstensi. Awalnya, daftar `.kitchen.yml` run berisi resep default buku masak,`installpkg::default`. Itulah resep yang akan Anda terapkan, jadi Anda tidak perlu memodifikasi daftar run.

5. Buat subdirektori `installpkg` bernama`recipes`.

   Jika buku masak berisi resep—kebanyakan melakukannya—mereka harus ada di subdirektori. `recipes`

Anda sekarang dapat menambahkan resep ke buku masak dan menggunakan Test Kitchen untuk menjalankannya pada sebuah instance.

Untuk menjalankan resep

1. Buat file bernama `default.rb` yang berisi kode contoh instalasi Git dari awal bagian dan simpan ke `recipes` subdirektori.
2. Di `installpkg` direktori, jalankan`kitchen converge`. Perintah ini memulai instance Ubuntu baru di Vagrant, menyalin buku masak Anda ke instance, dan memulai Chef run untuk mengeksekusi resep dalam daftar run. `.kitchen.yml`
3. Untuk memverifikasi bahwa resep berhasil, jalankan`kitchen login`, yang membuka koneksi SSH ke instance. Kemudian jalankan `git --version` untuk memverifikasi bahwa Git berhasil diinstal. Untuk kembali ke workstation Anda, jalankan`exit`.
4. Setelah selesai, jalankan `kitchen destroy` untuk mematikan instance. Contoh selanjutnya menggunakan buku masak yang berbeda.

Contoh ini adalah cara yang baik untuk memulai, tetapi sangat sederhana. Paket lain bisa lebih rumit untuk diinstal; Anda mungkin perlu melakukan salah satu atau semua hal berikut:

- Buat dan konfigurasikan pengguna.
- Buat satu atau lebih direktori untuk data, log, dan sebagainya.
- Instal satu atau lebih file konfigurasi.
- Tentukan nama paket atau nilai atribut yang berbeda untuk sistem operasi yang berbeda.
- Mulai layanan dan kemudian restart sesuai kebutuhan.

Contoh berikut menjelaskan cara mengatasi masalah ini, bersama dengan beberapa operasi berguna lainnya.

Contoh 2: Mengelola Pengguna

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tugas sederhana lainnya adalah mengelola pengguna pada sebuah instance. Resep berikut menambahkan pengguna baru ke instance Linux.

```
user "myuser" do
  home "/home/newuser"
  shell "/bin/bash"
end
```

Anda menggunakan sumber daya pengguna untuk mengelola pengguna di sistem Linux dan Windows, meskipun beberapa atribut hanya berlaku untuk satu sistem. Contoh membuat pengguna bernama `myuser` dan menentukan direktori home dan shell mereka. Tidak ada tindakan yang ditentukan, sehingga sumber daya menggunakan `create` tindakan default. Anda dapat menambahkan atribut `user` untuk menentukan berbagai pengaturan lain, seperti kata sandi atau ID grup. Anda juga dapat menggunakan `user` untuk tugas-tugas manajemen pengguna terkait seperti memodifikasi pengaturan pengguna atau menghapus pengguna. Untuk informasi lebih lanjut, lihat pengguna.

Untuk menjalankan resep

1. Buat direktori dalam `opsworks_cookbooks` bernama `newuser` dan navigasikan ke sana.
2. Buat `metadata.rb` file yang berisi kode berikut dan simpan ke`newuser`.

   ```
   name "newuser"
   version "0.1.0"
   ```

3. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket, dan tambahkan `recipes` direktori di dalam `newuser` direktori.
4. Tambahkan `default.rb` file dengan contoh resep ke `recipes` direktori buku masak.
5. Jalankan `kitchen converge` untuk mengeksekusi resep.
6. Gunakan `kitchen login` untuk masuk ke instance dan memverifikasi keberadaan pengguna baru dengan menjalankan`cat /etc/passwd`. `myuser`Pengguna harus berada di bagian bawah file.

Contoh 3: Membuat Direktori

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Ketika Anda menginstal paket pada sebuah instance, Anda sering perlu membuat beberapa file konfigurasi dan menempatkannya di direktori yang sesuai. Namun, direktori tersebut mungkin belum ada. Anda mungkin juga perlu membuat direktori untuk data, file log, dan sebagainya. Misalnya, Anda pertama kali mem-boot sistem Ubuntu yang Anda gunakan untuk sebagian besar contoh, `/srv` direktori tidak memiliki subdirektori. Jika Anda menginstal server aplikasi, Anda mungkin menginginkan `/srv/www/` direktori dan mungkin beberapa subdirektori untuk file data, log, dan sebagainya. Resep berikut dibuat `/srv/www/` pada sebuah instance.

```
directory "/srv/www/" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

Anda menggunakan `directory`sumber daya untuk membuat dan mengkonfigurasi direktori pada sistem Linux dan Windows, meskipun beberapa atribut digunakan secara berbeda. Nama sumber daya adalah nilai default untuk `path` atribut sumber daya, sehingga contoh membuat `/srv/www/` dan menentukan`mode`,`owner`, dan `group` propertinya.

Untuk menjalankan resep

1. Buat direktori di dalam `opsworks_cookbooks` bernama `createdir` dan arahkan ke sana.
2. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket, dan tambahkan `recipes` direktori di dalamnya`createdir`.
3. Tambahkan `default.rb` file dengan kode resep ke `recipes` subdirektori buku masak.

4. Jalankan `kitchen converge` untuk mengeksekusi resep.
5. Jalankan`kitchen login`, navigasikan ke `/srv` dan verifikasi bahwa ia memiliki `www` subdirektori.
6. Jalankan `exit` untuk kembali ke workstation Anda tetapi biarkan instance berjalan.

> **Note**
>
> Untuk membuat direktori relatif terhadap direktori home Anda pada instance, gunakan `#{ENV['HOME']}` untuk mewakili direktori home. Misalnya, berikut ini membuat `~/shared` direktori.
>
> ```
> directory "#{ENV['HOME']}/shared" do
>   ...
> end
> ```

Misalkan Anda ingin membuat direktori bersarang yang lebih dalam, seperti`/srv/www/shared`. Anda dapat memodifikasi resep sebelumnya sebagai berikut.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  action :create
end
```

Untuk menjalankan resep

1. Ganti kode `default.rb` dengan resep sebelumnya.
2. Jalankan `kitchen converge` dari `createdir` direktori.
3. Untuk memverifikasi bahwa direktori memang dibuat, jalankan`kitchen login`, navigasikan ke`/srv/www`, dan verifikasi bahwa itu berisi `shared` subdirektori.
4. `kitchen destroy`Jalankan untuk mematikan instance.

Anda akan melihat `kitchen converge` perintah berjalan lebih cepat. Itu karena instance sudah berjalan, jadi tidak perlu mem-boot instance, menginstal Chef, dan sebagainya. Uji Dapur hanya untuk menyalin buku masak yang diperbarui ke instance dan memulai menjalankan Chef.

Sekarang jalankan `kitchen converge` lagi, yang mengeksekusi resep pada instance baru. Anda sekarang akan melihat hasil berikut.

```
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]
>>>>>> ----------------------
```

Apa yang terjadi? Masalahnya adalah bahwa secara default, `directory` sumber daya hanya dapat membuat satu direktori pada satu waktu; itu tidak dapat membuat rantai direktori. Alasan resep bekerja sebelumnya adalah bahwa resep pertama yang Anda jalankan pada instance telah dibuat`/srv/www`, jadi membuat `/srv/www/shared` hanya satu subdirektori.

> **Note**
>
> Ketika Anda menjalankan`kitchen converge`, pastikan Anda tahu apakah Anda menjalankan resep Anda pada instance baru atau yang sudah ada. Anda mungkin mendapatkan hasil yang berbeda.

Untuk membuat rantai subdirektori, tambahkan `recursive` atribut ke `directory` dan atur ke. `true` Resep berikut dibuat `/srv/www/shared` langsung pada contoh yang bersih.

```
directory "/srv/www/shared" do
  mode 0755
```

```
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

Contoh 4: Menambahkan Kontrol Aliran

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Beberapa resep hanyalah serangkaian sumber daya Chef. Dalam hal ini, ketika Anda menjalankan resep, itu hanya mengeksekusi masing-masing penyedia sumber daya secara berurutan. Namun, seringkali berguna untuk memiliki jalur eksekusi yang lebih canggih. Berikut ini adalah dua skenario umum:

- Anda ingin resep untuk mengeksekusi sumber daya yang sama beberapa kali dengan pengaturan atribut yang berbeda.
- Anda ingin menggunakan pengaturan atribut yang berbeda pada sistem operasi yang berbeda.

Anda dapat mengatasi skenario seperti ini dengan memasukkan struktur kontrol Ruby ke dalam resep. Bagian ini menunjukkan cara memodifikasi resep dari Contoh 3: Membuat Direktori untuk mengatasi kedua skenario.

Topik

- Iterasi
- Logika Bersyarat

Iterasi

Contoh 3: Membuat Direktorimenunjukkan cara menggunakan `directory` sumber daya untuk membuat direktori atau rantai direktori. Namun, anggaplah Anda ingin membuat dua direktori

terpisah, `/srv/www/config` dan`/srv/www/shared`. Anda dapat menerapkan sumber daya direktori terpisah untuk setiap direktori, tetapi pendekatan itu bisa menjadi rumit jika Anda ingin membuat sangat banyak direktori. Resep berikut menunjukkan cara yang lebih sederhana untuk menangani tugas.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
end
```

Alih-alih menggunakan sumber daya direktori terpisah untuk setiap subdirektori, resep menggunakan koleksi string yang berisi jalur subdirektori. `each`Metode Ruby mengeksekusi sumber daya sekali untuk setiap elemen koleksi, dimulai dengan yang pertama. Nilai elemen diwakili dalam sumber daya oleh `path` variabel, yang dalam hal ini mewakili jalur direktori. Anda dapat dengan mudah mengadaptasi contoh ini untuk membuat sejumlah subdirektori.

Untuk menjalankan resep

1. Tetap di `createdir` direktori; Anda akan menggunakan buku masak itu untuk beberapa contoh berikutnya.
2. Jika Anda belum melakukannya, jalankan `kitchen destroy` sehingga Anda memulai dengan instance yang bersih.
3. Ganti kode `default.rb` dengan contoh dan jalankan`kitchen converge`.
4. Masuk ke instance; Anda akan melihat direktori yang baru dibuat di bawah`/srv`.

Anda dapat menggunakan tabel hash untuk menentukan dua nilai untuk setiap iterasi. Resep berikut menciptakan `/srv/www/config` dan`/srv/www/shared`, masing-masing dengan mode yang berbeda.

```
{ "/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
```

Untuk menjalankan resep

1. Jika Anda belum melakukannya, jalankan `kitchen destroy` sehingga Anda memulai dengan instance yang bersih.
2. Ganti kode `default.rb` dengan contoh dan jalankan`kitchen converge`.
3. Masuk ke instance; Anda akan melihat direktori yang baru dibuat di bawah `/srv` dengan mode yang ditentukan.

> **Note**
>
> AWS OpsWorks Resep tumpukan biasanya menggunakan pendekatan ini untuk mengekstrak nilai dari konfigurasi tumpukan dan penerapan JSON —yang pada dasarnya adalah tabel hash besar—dan memasukkannya ke dalam sumber daya. Sebagai contoh, lihat Menyebarkan Resep.

Logika Bersyarat

Anda juga dapat menggunakan logika bersyarat Ruby untuk membuat beberapa cabang eksekusi. Resep berikut menggunakan `if-elsif-else` logika untuk memperluas contoh sebelumnya sehingga menciptakan subdirektori bernama`/srv/www/shared`, tetapi hanya pada sistem Debian dan Ubuntu. Untuk semua sistem lain, ia mencatat pesan kesalahan yang ditampilkan di output 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
```

Untuk menjalankan contoh resep

1. Jika instance Anda masih aktif, jalankan `kitchen destroy` untuk mematikannya.
2. Ganti kode `default.rb` dengan kode contoh.
3. Edit `.kitchen.yml` untuk menambahkan sistem CentOS 6.4 ke daftar platform. `platforms`Bagian file sekarang akan terlihat seperti.

   ```
   ...
   platforms:
     - name: ubuntu-12.04
     - name: centos-6.4
   ...
   ```

4. Jalankan`kitchen converge`, yang akan membuat instance dan menjalankan resep untuk setiap platform `.kitchen.yml` secara berurutan.

   > **Note**
   >
   > Jika Anda ingin menyatu hanya satu contoh, tambahkan nama instance sebagai parameter. Misalnya, untuk menyatukan resep hanya pada platform Ubuntu, jalankan`kitchen converge default-ubuntu-1204`. Jika Anda lupa nama platform, jalankan saja`kitchen list`.

Anda akan melihat pesan log Anda di bagian CentOS dari output Test Kitchen, yang akan terlihat seperti berikut:

```
...
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
```

Anda sekarang dapat masuk ke instance dan memverifikasi bahwa direktori telah atau tidak dibuat. Namun, Anda tidak bisa lari begitu saja `kitchen login` sekarang. Anda harus menentukan instance mana dengan menambahkan nama platform, misalnya,`kitchen login default-ubuntu-1204`.

> **Note**
>
> Jika perintah Test Kitchen mengambil nama instance, Anda tidak perlu mengetikkan nama lengkapnya. Test Kitchen memperlakukan nama instance sebagai ekspresi reguler Ruby, jadi Anda hanya perlu karakter yang cukup untuk memberikan kecocokan yang unik. Misalnya, Anda dapat menyatukan hanya instance Ubuntu dengan menjalankan `kitchen converge ub` atau masuk ke instance CentOS dengan menjalankan. `kitchen login 64`

Pertanyaan yang mungkin Anda miliki pada titik ini adalah bagaimana resep mengetahui platform mana yang sedang berjalan. Chef menjalankan alat yang disebut Ohai untuk setiap proses yang mengumpulkan data sistem, termasuk platform, dan mewakilinya sebagai satu set atribut dalam struktur yang disebut objek node. `platform`?Metode Chef membandingkan sistem dalam tanda kurung dengan nilai platform Ohai, dan mengembalikan nilai true jika salah satunya cocok.

Anda dapat mereferensikan nilai atribut node langsung dalam kode Anda dengan menggunakan`node['attribute_name']`. Nilai platform, misalnya, diwakili oleh`node['platform']`. Anda dapat, misalnya, telah menulis contoh sebelumnya sebagai berikut.

```
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
```

Alasan umum untuk memasukkan logika bersyarat dalam resep adalah untuk mengakomodasi fakta bahwa keluarga Linux yang berbeda terkadang menggunakan nama yang berbeda untuk paket, direktori, dan sebagainya. Misalnya, nama paket Apache ada `httpd` di sistem CentOS `apache2` dan pada sistem Ubuntu.

Jika Anda hanya membutuhkan string yang berbeda untuk sistem yang berbeda, `value_for_platform`metode Chef adalah solusi yang lebih sederhana daripada`if-elsif-else`. Resep berikut membuat `/srv/www/shared` direktori pada sistem CentOS, `/srv/www/data` direktori pada sistem Ubuntu, dan yang `/srv/www/config` lainnya.

```
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`menetapkan jalur yang sesuai ke `data_dir` dan `directory` sumber daya menggunakan nilai itu untuk membuat direktori.

Untuk menjalankan contoh resep

1. Jika instance Anda masih aktif, jalankan `kitchen destroy` untuk mematikannya.
2. Ganti kode `default.rb` dengan kode contoh.
3. Jalankan `kitchen converge` dan kemudian masuk ke setiap instance untuk memverifikasi bahwa direktori yang sesuai ada.

Contoh 5: Menggunakan Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Resep di bagian sebelumnya menggunakan nilai hard-code untuk segala sesuatu selain platform. Pendekatan ini bisa merepotkan jika, misalnya, Anda ingin menggunakan nilai yang sama di lebih dari satu resep. Anda dapat menentukan nilai secara terpisah dari resep dengan menyertakan file atribut di buku masak Anda.

File atribut adalah aplikasi Ruby yang memberikan nilai ke satu atau lebih atribut. Itu harus ada di `attributes` folder buku masak. Chef menggabungkan atribut ke dalam objek node dan resep apa pun dapat menggunakan nilai atribut dengan mereferensikan atribut. Topik ini menunjukkan cara memodifikasi resep dari Iterasi untuk menggunakan atribut. Inilah resep asli untuk referensi.

```
[ "/srv/www/config", "/srv/www/shared" ].each do |path|
  directory path do
    mode 0755
    owner 'root'
    group 'root'
    recursive true
    action :create
  end
end
```

Berikut ini mendefinisikan atribut untuk nama subdirektori, mode, pemilik, dan nilai grup.

```
default['createdir']['shared_dir'] = 'shared'
default['createdir']['config_dir'] = 'config'
default['createdir']['mode'] = 0755
default['createdir']['owner'] = 'root'
default['createdir']['group'] = 'root'
```

Perhatikan hal berikut:

- Setiap definisi dimulai dengan tipe atribut.

  Jika atribut didefinisikan lebih dari sekali — mungkin dalam file atribut yang berbeda — tipe atribut menentukan prioritas atribut, yang menentukan definisi mana yang dimasukkan ke dalam objek

node. Untuk informasi selengkapnya, lihat Atribut Prioritas. Semua definisi dalam contoh ini memiliki tipe `default` atribut, yang merupakan tipe biasa untuk tujuan ini.

- Atribut memiliki nama bersarang.

  Objek node pada dasarnya adalah tabel hash yang dapat bersarang secara sewenang-wenang, sehingga nama atribut dapat dan umumnya bersarang. File atribut ini mengikuti praktik standar menggunakan nama bersarang dengan nama buku masak,`createdir`, sebagai elemen pertama.

Alasan untuk menggunakan createdir sebagai elemen pertama atribut adalah ketika Anda menjalankan Chef, Chef menggabungkan atribut dari setiap buku masak ke dalam objek node. Dengan AWS OpsWorks Stacks, objek node menyertakan sejumlah besar atribut dari buku masak bawaan selain atribut apa pun yang Anda tentukan. Menyertakan nama buku masak dalam nama atribut mengurangi risiko tabrakan nama dengan atribut dari buku masak lain, terutama jika atribut Anda memiliki nama seperti atau. `port` `user` Jangan beri nama atribut seperti `[:apache2]` `[:user]`, misalnya, kecuali jika Anda ingin mengganti nilai atribut itu. Untuk informasi selengkapnya, lihat Menggunakan Atribut Buku Masak Kustom.

Contoh berikut menunjukkan resep asli menggunakan atribut bukan nilai hard-code.

```
[ "/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**
>
> Jika Anda ingin memasukkan nilai atribut ke dalam string, bungkus dengan`#{}`. Pada contoh sebelumnya, `#{node['createdir']['shared_dir']}` tambahkan “shared” ke “/srv/www/”.

Untuk menjalankan resep

1. Jalankan `kitchen destroy` untuk memulai dengan instance bersih.
2. Ganti kode `recipes/default.rb` dengan contoh resep sebelumnya.
3. Buat subdirektori `createdir` bernama `attributes` dan tambahkan file bernama `default.rb` yang berisi definisi atribut.
4. Edit `.kitchen.yml` untuk menghapus CentOS dari daftar platform.
5. Jalankan `kitchen converge` dan kemudian masuk ke instance dan verifikasi itu `/srv/www/shared` dan `/srv/www/config` ada di sana.

> **Note**
>
> Dengan AWS OpsWorks Stacks, mendefinisikan nilai sebagai atribut memberikan manfaat tambahan; Anda dapat menggunakan JSON khusus untuk mengganti nilai tersebut berdasarkan per-stack atau bahkan per-deployment. Ini dapat berguna untuk berbagai tujuan, termasuk yang berikut:
>
> - Anda dapat menyesuaikan perilaku resep Anda, seperti pengaturan konfigurasi atau nama pengguna, tanpa harus mengubah buku masak.
>
> Anda dapat, misalnya, menggunakan buku masak yang sama untuk tumpukan yang berbeda dan menggunakan JSON khusus untuk menentukan pengaturan konfigurasi kunci untuk tumpukan tertentu. Ini menghemat waktu dan upaya yang diperlukan untuk memodifikasi buku masak atau menggunakan buku masak yang berbeda untuk setiap tumpukan.
> - Anda tidak perlu memasukkan informasi yang berpotensi sensitif seperti kata sandi basis data di repositori buku masak Anda.
>
> Sebagai gantinya, Anda dapat menggunakan atribut untuk menentukan nilai default dan kemudian menggunakan JSON khusus untuk mengganti nilai itu dengan yang asli.
>
> Untuk informasi selengkapnya tentang cara menggunakan JSON kustom untuk mengganti atribut, lihat. Mengesampingkan Atribut

File atribut dinamai `default.rb` karena merupakan aplikasi Ruby, jika agak sederhana. Itu berarti Anda dapat, misalnya, menggunakan logika bersyarat untuk menentukan nilai atribut berdasarkan sistem operasi. DiLogika Bersyarat, Anda menentukan nama subdirektori yang berbeda untuk keluarga Linux yang berbeda dalam resep. Dengan file atribut, Anda dapat menempatkan logika kondisional dalam file atribut.

File atribut berikut menggunakan `value_for_platform` untuk menentukan nilai `['shared_dir']` atribut yang berbeda, tergantung pada sistem operasi. Untuk kondisi lain, Anda dapat menggunakan `if-elsif-else` logika Ruby atau `case` pernyataan.

```
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'
```

Untuk menjalankan resep

1. Jalankan `kitchen destroy` untuk memulai dengan contoh baru.
2. Ganti kode `attributes/default.rb` dengan contoh sebelumnya.
3. Edit `.kitchen.yml` untuk menambahkan platform CentOS ke bagian platform, seperti yang dijelaskan dalam. Logika Bersyarat
4. Jalankan`kitchen converge`, lalu masuk ke instance untuk memverifikasi bahwa direktori ada di sana.

Setelah selesai, jalankan `kitchen destroy` untuk menghentikan instance. Contoh selanjutnya menggunakan buku masak baru.

Contoh 6: Membuat File

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda membuat direktori, Anda sering perlu mengisinya dengan file konfigurasi, file data, dan sebagainya. Topik ini menunjukkan dua cara untuk menginstal file pada sebuah instance.

Topik

- Menginstal File dari Cookbook
- Membuat File dari Template

Menginstal File dari Cookbook

Cara termudah untuk menginstal file pada instance adalah dengan menggunakan `cookbook_file`sumber daya, yang menyalin file dari buku masak ke lokasi tertentu pada instance untuk sistem Linux dan Windows. Contoh ini memperluas resep dari Contoh 3: Membuat Direktori untuk menambahkan file data `/srv/www/shared` setelah direktori dibuat. Sebagai referensi, berikut resep aslinya.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

Untuk mengatur buku masak

1. Di dalam `opsworks_cookbooks` direktori, buat direktori bernama `createfile` dan navigasikan ke sana.

2. Tambahkan `metadata.rb` file `createfile` dengan konten berikut.

```
name "createfile"
version "0.1.0"
```

3. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket, dan hapus CentOS dari `platforms` daftar.
4. Tambahkan `recipes` subdirektori ke`createfile`.

File yang akan diinstal berisi data JSON berikut.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true
}
```

Untuk mengatur file data

1. Tambahkan `files` subdirektori ke `createfile` dan `default` subdirektori ke. `files` Setiap file yang Anda instal dengan `cookbook_file` harus dalam subdirektori`files`, seperti `files/default` dalam contoh ini.

> **Note**
>
> Jika Anda ingin menentukan file yang berbeda untuk sistem yang berbeda, Anda dapat menempatkan setiap file khusus sistem dalam subfolder yang diberi nama untuk sistem, seperti. `files/ubuntu` `cookbook_file`Sumber daya menyalin file khusus sistem yang sesuai, jika ada, dan sebaliknya menggunakan file tersebut. `default` Untuk informasi selengkapnya, lihat cookbook_file.

2. Buat file bernama `example_data.json` dengan JSON dari contoh sebelumnya dan tambahkan ke. `files/default`

Resep berikut menyalin `example_data.json` ke lokasi tertentu.

```
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
```

Setelah sumber daya direktori dibuat`/srv/www/shared`, `cookbook_file` sumber daya menyalin `example_data.json` ke direktori itu dan juga menetapkan pengguna, grup, dan mode file.

> **Note**
>
> Sumber `cookbook_file` daya memperkenalkan tindakan baru:`create_if_missing`. Anda juga dapat menggunakan `create` tindakan, tetapi itu menimpa file yang ada. Jika Anda tidak ingin menimpa apa pun, gunakan`create_if_missing`, yang menginstal `example_data.json` hanya jika belum ada.

Untuk menjalankan resep

1. Jalankan `kitchen destroy` untuk memulai dengan contoh baru.
2. Buat `default.rb` file yang berisi resep sebelumnya dan simpan ke. `recipes`
3. Jalankan`kitchen converge`, lalu masuk ke instance untuk memverifikasi yang `/srv/www/shared` berisi`example_data.json`.

Membuat File dari Template

`cookbook_file`Sumber daya ini berguna untuk beberapa tujuan, tetapi hanya menginstal file apa pun yang Anda miliki di buku masak. `template`Sumber daya menyediakan cara yang lebih fleksibel untuk menginstal file pada instance Windows atau Linux dengan membuatnya secara dinamis dari template. Anda kemudian dapat menentukan detail konten file saat runtime dan mengubahnya sesuai

kebutuhan. Misalnya, Anda mungkin ingin file konfigurasi memiliki pengaturan tertentu ketika Anda memulai instance dan memodifikasi pengaturan nanti ketika Anda menambahkan lebih banyak instance ke tumpukan.

Contoh ini memodifikasi `createfile` buku masak untuk menggunakan `template` sumber daya untuk menginstal versi yang sedikit dimodifikasi. `example_data.json`

Inilah yang akan terlihat seperti file yang diinstal.

```
{
  "my_name" : "myname",
  "your_name" : "yourname",
  "a_number" : 42,
  "a_boolean" : true,
  "a_string" : "some string",
  "platform" : "ubuntu"
}
```

Sumber daya template biasanya digunakan bersama dengan file atribut, jadi contoh menggunakan satu untuk menentukan nilai-nilai berikut.

```
default['createfile']['my_name'] = 'myname'
default['createfile']['your_name'] = 'yourname'
default['createfile']['install_file'] = true
```

Untuk mengatur buku masak

1. Hapus `files` direktori `createfile` buku masak dan isinya.
2. Tambahkan `attributes` subdirektori ke `createfile` dan tambahkan `default.rb` file `attributes` yang berisi definisi atribut sebelumnya.

Template adalah `.erb` file yang pada dasarnya adalah salinan dari file akhir, dengan beberapa konten diwakili oleh placeholder. Saat `template` sumber daya membuat file, ia menyalin konten template ke file yang ditentukan, dan menimpa placeholder dengan nilai yang ditetapkan. Berikut template untuk`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'] %>"
}
```

`<%=...%>`Nilainya adalah placeholder.

- `<%=node[...]%>`merupakan nilai atribut node.

  Untuk contoh ini, nilai “your_name” adalah placeholder yang mewakili salah satu nilai atribut dari file atribut buku masak.
- `<%=@...%>`mewakili nilai variabel yang didefinisikan dalam sumber daya template, seperti yang dibahas segera.

Untuk membuat file template

1. Tambahkan `templates` subdirektori ke `createfile` buku masak dan `default` subdirektori ke. `templates`

   > **Note**
   >
   > `templates`Direktori bekerja seperti `files` direktori. Anda dapat menempatkan template khusus sistem dalam subdirektori seperti `ubuntu` yang dinamai untuk sistem. `template`Sumber daya menggunakan templat khusus sistem yang sesuai jika ada dan menggunakan templat. `default`
2. Buat file bernama `example_data.json.erb` dan masukkan ke dalam `templates/default` direktori. Nama template bersifat arbitrer, tetapi Anda biasanya membuatnya dengan menambahkan `.erb` ke nama file, termasuk ekstensi apa pun.

Resep berikut menggunakan sumber `template` daya untuk membuat`/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
```

Sumber `template` daya dibuat `example_data.json` dari template dan menginstalnya di`/srv/www/shared`.

- Nama template,`/srv/www/shared/example_data.json`, menentukan jalur dan nama file yang diinstal.
- `source`Atribut menentukan template yang digunakan untuk membuat file.
- `mode`Atribut menentukan mode file yang diinstal.
- Sumber daya mendefinisikan dua variabel, `a_boolean_var` dan`a_string_var`.

  Ketika sumber daya dibuat`example_data.json`, ia menimpa placeholder variabel dalam template dengan nilai yang sesuai dari sumber daya.
- Atribut `only_if` guard mengarahkan sumber daya untuk membuat file hanya jika `['createfile']['install_file']` diatur ke`true`.

Untuk menjalankan resep

1. Jalankan `kitchen destroy` untuk memulai dengan contoh baru.
2. Ganti kode `recipes/default.rb` dengan contoh sebelumnya.
3. Jalankan`kitchen converge`, lalu masuk ke instance untuk memverifikasi bahwa file tersebut masuk `/srv/www/shared` dan memiliki konten yang benar.

Setelah selesai, jalankan `kitchen destroy` untuk mematikan instance. Bagian selanjutnya menggunakan buku masak baru.

Contoh 7: Menjalankan Perintah dan Skrip

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sumber daya koki dapat menangani berbagai macam tugas pada sebuah instance, tetapi terkadang lebih baik menggunakan perintah shell atau skrip. Misalnya, Anda mungkin sudah memiliki skrip yang Anda gunakan untuk menyelesaikan tugas tertentu, dan akan lebih mudah untuk terus menggunakannya daripada menerapkan kode baru. Bagian ini menunjukkan cara menjalankan perintah atau skrip pada sebuah instance.

Topik

- Menjalankan Perintah
- Menjalankan Skrip

Menjalankan Perintah

Sumber `script`daya menjalankan satu atau lebih perintah. Ini mendukung penerjemah perintah csh, bash, Perl, Python, dan Ruby, sehingga dapat digunakan pada sistem Linux atau Windows selama mereka memiliki interpreter yang sesuai diinstal. Topik ini menunjukkan cara menjalankan perintah bash sederhana pada instance Linux. Chef juga mendukung powershell_script dan sumber daya batch untuk menjalankan skrip di Windows. Untuk informasi selengkapnya, lihat Menjalankan PowerShell Skrip Windows.

Untuk memulai

1. Di dalam `opsworks_cookbooks` direktori, buat direktori bernama `script` dan navigasikan ke sana.
2. Tambahkan `metadata.rb` file `script` dengan konten berikut.

```
name "script"
version "0.1.0"
```

3. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket, dan hapus CentOS dari `platforms` daftar.
4. Di dalam`script`, buat direktori bernama`recipes`.

Anda dapat menjalankan perintah dengan menggunakan `script` sumber daya itu sendiri, tetapi Chef juga mendukung satu set versi khusus penerjemah perintah dari sumber daya, yang dinamai untuk penerjemah. Resep berikut menggunakan `bash`sumber daya untuk menjalankan skrip bash sederhana.

```
bash "install_something" do
  user "root"
  cwd "/tmp"
  code <<-EOH
    touch somefile
  EOH
  not_if do
    File.exists?("/tmp/somefile")
  end
end
```

Sumber `bash` daya dikonfigurasi sebagai berikut.

- Ini menggunakan tindakan default,`run`, yang menjalankan perintah di `code` blok.

  Contoh ini memiliki satu perintah`touch somefile`, tetapi `code` blok dapat berisi beberapa perintah.
- `user`Atribut menentukan pengguna yang mengeksekusi perintah.
- `cwd`Atribut menentukan direktori kerja.

  Untuk contoh ini, `touch` buat file di `/tmp` direktori.
- Atribut `not_if` guard mengarahkan sumber daya untuk tidak mengambil tindakan jika file sudah ada.

Untuk menjalankan resep

1. Buat `default.rb` file yang berisi kode contoh sebelumnya dan simpan ke. `recipes`
2. Jalankan`kitchen converge`, lalu masuk ke instance untuk memverifikasi bahwa file tersebut masuk`/tmp`.

Menjalankan Skrip

`script`Sumber daya nyaman, terutama jika Anda hanya perlu menjalankan satu atau dua perintah, tetapi seringkali lebih baik untuk menyimpan skrip dalam file dan menjalankan file. `execute`Sumber daya menjalankan file yang dapat dieksekusi tertentu, termasuk file skrip, di Linux atau Windows. Topik ini memodifikasi `script` buku masak dari contoh sebelumnya untuk digunakan `execute` untuk menjalankan skrip shell sederhana. Anda dapat dengan mudah memperluas contoh ke skrip yang lebih kompleks, atau jenis file yang dapat dieksekusi lainnya.

Untuk mengatur file skrip

1. Tambahkan `files` subdirektori ke `script` dan `default` subdirektori ke. `files`
2. Buat file bernama `touchfile` yang berisi berikut ini dan tambahkan ke`files/default`. Baris interpreter Bash umum digunakan dalam contoh ini, tetapi gantikan interpreter yang berfungsi untuk lingkungan shell Anda jika perlu.

   ```
   #!/usr/bin/env bash
   touch somefile
   ```

   File skrip dapat berisi sejumlah perintah. Untuk kenyamanan, contoh skrip ini hanya memiliki satu `touch` perintah.

Resep berikut mengeksekusi skrip.

```
cookbook_file "/tmp/touchfile" do
  source "touchfile"
  mode 0755
end

execute "touchfile" do
  user "root"
  cwd "/tmp"
```

```
  command "./touchfile"
end
```

`cookbook_file`Sumber daya menyalin file skrip ke `/tmp` dan mengatur mode untuk membuat file dapat dieksekusi. Sumber `execute` daya kemudian mengeksekusi file sebagai berikut:

- `user`Atribut menentukan pengguna perintah (`root`dalam contoh ini).
- `cwd`Atribut menentukan direktori kerja (`/tmp`dalam contoh ini).
- `command`Atribut menentukan skrip yang akan dieksekusi (`touchfile`dalam contoh ini), yang terletak di direktori kerja.

Untuk menjalankan resep

1. Ganti kode `recipes/default.rb` dengan contoh sebelumnya.
2. Jalankan`kitchen converge`, lalu masuk ke instance untuk memverifikasi bahwa `/tmp` sekarang berisi file skrip, dengan mode diatur ke 0755, dan`somefile`.

Setelah selesai, jalankan `kitchen destroy` untuk mematikan instance. Bagian selanjutnya menggunakan buku masak baru.

Contoh 8: Mengelola Layanan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Paket seperti server aplikasi biasanya memiliki layanan terkait yang harus dimulai, dihentikan, dimulai ulang, dan sebagainya. Misalnya, Anda perlu memulai layanan Tomcat setelah menginstal paket atau setelah instance selesai booting, dan restart layanan setiap kali Anda memodifikasi file konfigurasi. Topik ini membahas dasar-dasar bagaimana mengelola layanan pada instance Linux, menggunakan server aplikasi Tomcat sebagai contoh. Sumber daya layanan bekerja dengan cara yang sama pada

instance Windows, meskipun ada beberapa perbedaan secara detail. Untuk informasi selengkapnya, lihat `service`.

> **Note**
>
> Contohnya melakukan instalasi Tomcat yang sangat minim, cukup untuk menunjukkan dasar-dasar cara menggunakan `service` sumber daya. Untuk contoh bagaimana menerapkan resep untuk server Tomcat yang lebih fungsional, lihatMembuat Layer Server Tomcat Kustom.

Topik

- Mendefinisikan dan Memulai Layanan
- Menggunakan pemberitahuan untuk Memulai atau Memulai Ulang Layanan

Mendefinisikan dan Memulai Layanan

Bagian ini menunjukkan dasar-dasar cara mendefinisikan dan memulai layanan.

Untuk memulai

1. Di `opsworks_cookbooks` direktori, buat direktori bernama `tomcat` dan navigasikan ke sana.
2. Tambahkan `metadata.rb` file `tomcat` dengan konten berikut.

   ```
   name "tomcat"
   version "0.1.0"
   ```
3. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket, dan hapus CentOS dari `platforms` daftar.
4. Tambahkan `recipes` subdirektori ke`tomcat`.

Anda menggunakan `service`sumber daya untuk mengelola layanan. Resep default berikut menginstal Tomcat dan memulai layanan.

```
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
```

Resepnya melakukan hal berikut:

- Sumber `execute` daya berjalan `apt-get update` untuk menginstal pembaruan sistem saat ini.

  Untuk contoh Ubuntu yang digunakan dalam contoh ini, Anda harus menginstal pembaruan sebelum menginstal Tomcat. Sistem lain mungkin memiliki persyaratan yang berbeda.
- Sumber `package` daya menginstal Tomcat 7.
- `tomcat::service`Resep yang disertakan mendefinisikan layanan dan dibahas nanti.
- Sumber `service` daya memulai layanan Tomcat.

  Anda juga dapat menggunakan sumber daya ini untuk mengeluarkan perintah lain, seperti menghentikan dan memulai ulang layanan.

Contoh berikut menunjukkan `tomcat::service` resepnya.

```
service 'tomcat' do
  service_name "tomcat7"
  supports :restart => true, :reload => false, :status => true
  action :nothing
end
```

Resep ini menciptakan definisi layanan Tomcat sebagai berikut:

- Nama sumber dayatomcat,, digunakan oleh resep lain untuk referensi layanan.

  Misalnya, `default.rb` referensi `tomcat` untuk memulai layanan.
- Sumber `service_name` daya menentukan nama layanan.

Saat Anda mencantumkan layanan pada instance, layanan Tomcat akan diberi nama tomcat7.

- `supports`menentukan bagaimana Chef mengelola layanan`restart`,`reload`, dan `status` perintah.
  - `true`menunjukkan bahwa Chef dapat menggunakan skrip init atau penyedia layanan lain untuk menjalankan perintah.
  - `false`menunjukkan bahwa Chef harus mencoba menjalankan perintah dengan cara lain.

Perhatikan bahwa `action` diatur ke`:nothing`, yang mengarahkan sumber daya untuk tidak mengambil tindakan. Sumber daya layanan mendukung tindakan seperti `start` dan`restart`. Namun, buku masak ini mengikuti praktik standar menggunakan definisi layanan yang tidak mengambil tindakan dan memulai atau memulai kembali layanan di tempat lain. Setiap resep yang memulai atau memulai ulang layanan harus terlebih dahulu mendefinisikannya, jadi pendekatan paling sederhana adalah menempatkan definisi layanan dalam resep terpisah dan memasukkannya ke dalam resep lain sesuai kebutuhan.

> **Note**
>
> Untuk mempermudah, resep default untuk contoh ini menggunakan `service` sumber daya untuk memulai layanan setelah menjalankan definisi layanan. Implementasi produksi biasanya memulai atau memulai ulang layanan dengan menggunakan`notifies`, seperti yang dibahas nanti.

Untuk menjalankan resep

1. Buat `default.rb` file yang berisi contoh resep default dan simpan ke`recipes`.
2. Buat `service.rb` file yang berisi contoh definisi layanan dan simpan ke`recipes`.
3. Jalankan`kitchen converge`, lalu masuk ke instance dan jalankan perintah berikut untuk memverifikasi bahwa layanan sedang berjalan.

   ```
   sudo service tomcat7 status
   ```

> **Note**
>
> Jika Anda menjalankan secara `service.rb` terpisah dari`default.rb`, Anda harus mengedit `.kitchen.yml` untuk menambahkan `tomcat::service` ke daftar run. Namun, ketika Anda memasukkan resep, kodenya dimasukkan ke dalam resep induk sebelum resep dijalankan. `service.rb`Oleh karena itu pada dasarnya merupakan bagian dari `default.rb` dan tidak memerlukan entri daftar lari terpisah.

Menggunakan pemberitahuan untuk Memulai atau Memulai Ulang Layanan

Implementasi produksi biasanya tidak digunakan `service` untuk memulai atau memulai ulang layanan. Sebaliknya, mereka `notifies` menambah salah satu dari beberapa sumber daya. Misalnya, jika Anda ingin memulai ulang layanan setelah memodifikasi file konfigurasi, Anda menyertakan `notifies` `template` sumber daya terkait. Menggunakan `notifies` memiliki keuntungan sebagai berikut dibandingkan menggunakan `service` sumber daya untuk memulai ulang layanan secara eksplisit.

- `notifies`Elemen memulai ulang layanan hanya jika file konfigurasi terkait telah berubah, sehingga tidak ada risiko menyebabkan restart layanan yang tidak perlu.
- Chef memulai ulang layanan paling banyak sekali di akhir setiap proses, terlepas dari berapa banyak `notifies` run yang dikandungnya.

  Misalnya, Chef run mungkin menyertakan beberapa sumber template, yang masing-masing memodifikasi file konfigurasi yang berbeda dan memerlukan layanan restart jika file telah berubah. Namun, Anda biasanya ingin memulai ulang layanan hanya sekali, di akhir menjalankan Chef. Jika tidak, Anda mungkin mencoba memulai ulang layanan yang belum sepenuhnya beroperasi dari restart sebelumnya, yang dapat menyebabkan kesalahan.

Contoh ini memodifikasi `tomcat::default` untuk menyertakan `template` sumber daya yang digunakan `notifies` untuk memulai ulang layanan. Contoh realistis akan menggunakan sumber daya template yang membuat versi khusus dari salah satu file konfigurasi Tomcat, tetapi itu agak panjang dan kompleks. Untuk kesederhanaan, contoh hanya menggunakan sumber daya template dariMembuat File dari Template. Ini tidak ada hubungannya dengan Tomcat, tetapi menyediakan cara sederhana untuk menunjukkan cara menggunakannya`notifies`. Untuk contoh cara menggunakan templat untuk membuat file konfigurasi Tomcat, lihatResep Pengaturan.

Untuk mengatur buku masak

1. Tambahkan `templates` subdirektori ke `tomcat` dan `default` subdirektori ke. `templates`
2. Salin `example_data.json.erb` template dari `createfile` buku masak ke `templates/default` direktori.
3. Tambahkan `attributes` subdirektori ke`tomcat`.
4. Salin file `default.rb` atribut dari `createfile` buku masak ke `attributes` direktori.

Resep berikut digunakan `notifies` untuk memulai kembali layanan 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
```

Contoh menggabungkan resep dari Membuat File dari Template ke dalam resep dari bagian sebelumnya, dengan dua perubahan signifikan:

- `service`Sumber daya masih ada, tetapi sekarang melayani tujuan yang agak berbeda.

  `:enable`Tindakan ini memungkinkan layanan Tomcat saat boot.
- Sumber daya template sekarang termasuk`notifies`, yang memulai ulang layanan Tomcat jika `example_data.json` telah berubah.

  Ini memastikan bahwa layanan dimulai ketika Tomcat pertama kali diinstal dan dimulai ulang setelah setiap perubahan konfigurasi.

Untuk menjalankan resep

1. Jalankan `kitchen destroy` untuk memulai dengan instance bersih.
2. Ganti kode `default.rb` dengan contoh sebelumnya.
3. Jalankan`kitchen converge`, lalu masuk ke instance dan verifikasi bahwa layanan sedang berjalan.

> **Note**
>
> Jika Anda ingin memulai ulang layanan tetapi resepnya tidak menyertakan sumber daya seperti `template` yang mendukung`notifies`, Anda dapat menggunakan `execute` sumber daya tiruan. Sebagai contoh
>
> ```
> execute 'trigger tomcat service restart' do
>   command 'bin/true'
>   notifies :restart, resources(:service => 'tomcat')
> end
> ```
>
> `execute`Sumber daya harus memiliki `command` atribut, bahkan jika Anda menggunakan sumber daya hanya sebagai cara untuk menjalankannya`notifies`. Contoh ini mengatasi

persyaratan itu dengan menjalankan`/bin/true`, yang merupakan perintah shell yang hanya mengembalikan kode sukses.

Contoh 9: Menggunakan EC2 Instans Amazon

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sampai titik ini, Anda telah menjalankan instance secara lokal di. VirtualBox Meskipun ini cepat dan mudah, Anda akhirnya ingin menguji resep Anda pada EC2 contoh Amazon. Secara khusus, jika Anda ingin menjalankan resep di Amazon Linux, itu hanya tersedia di Amazon EC2. Anda dapat menggunakan sistem serupa seperti CentOS untuk implementasi dan pengujian awal, tetapi satu-satunya cara untuk sepenuhnya menguji resep Anda di Amazon Linux adalah dengan instance Amazon EC2 .

Topik ini menunjukkan cara menjalankan resep pada EC2 instance Amazon. Anda akan menggunakan Test Kitchen dan Vagrant dengan cara yang sama seperti bagian sebelumnya, dengan dua perbedaan:

- Pengemudi itu `kitchen-ec2`bukan Vagrant.
- `.kitchen.yml`File buku masak harus dikonfigurasi dengan informasi yang diperlukan untuk meluncurkan EC2 instance Amazon.

> **Note**
>
> Pendekatan alternatif adalah dengan menggunakan plug-in `vagrant-aws` Vagrant. Untuk informasi selengkapnya, lihat Penyedia AWS Vagrant.

Anda memerlukan kredensyal AWS untuk membuat instans Amazon EC2 . Jika Anda tidak memiliki akun AWS, Anda dapat memperolehnya, sebagai berikut.

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Anda harus membuat pengguna IAM dengan izin untuk mengakses Amazon EC2 dan menyimpan akses pengguna dan kunci rahasia ke lokasi yang aman di workstation Anda. Test Kitchen akan menggunakan kredensyal tersebut untuk membuat instance. Cara yang lebih disukai untuk memberikan kredensil ke Test Kitchen adalah dengan menetapkan kunci ke variabel lingkungan berikut di workstation Anda.

> **Warning**
>
> Pengguna IAM memiliki kredensyal jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan bahwa Anda menghapus pengguna ini ketika mereka tidak lagi diperlukan.

- AWS_ACCESS_KEY — kunci akses pengguna Anda, yang akan terlihat sepertiAKIAIOSFODNN7EXAMPLE.
- AWS_SECRET_KEY — kunci rahasia pengguna Anda, yang akan terlihat sepertiwJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY.

Pendekatan ini mengurangi kemungkinan secara tidak sengaja membahayakan akun Anda dengan, misalnya, mengunggah proyek yang berisi kredensil Anda ke repositori publik.

Untuk mengatur buku masak

1. Untuk menggunakan `kitchen-ec2` driver, Anda harus memiliki `ruby-dev` paket yang diinstal pada sistem Anda. Contoh perintah berikut menunjukkan bagaimana menggunakan `aptitude` untuk menginstal paket pada sistem Ubuntu.

   ```
   sudo aptitude install ruby1.9.1-dev
   ```

2. `kitchen-ec2`Driver adalah permata, yang dapat Anda instal sebagai berikut:

   ```
   gem install kitchen-ec2
   ```

   Tergantung pada workstation Anda, perintah ini mungkin memerlukan`sudo`, atau Anda juga dapat menggunakan manajer lingkungan Ruby seperti RVM. Prosedur ini diuji dengan versi 0.8.0 `kitchen-ec2` driver, tetapi ada versi yang lebih baru. Untuk menginstal versi tertentu, jalankan`gem install kitchen-ec2 -v <version number>`.

3. Anda harus menentukan key pair Amazon EC2 SSH yang dapat digunakan Test Kitchen untuk menyambung ke instance. Jika Anda tidak memiliki EC2 key pair Amazon, lihat Amazon EC2 Key Pairs untuk informasi tentang cara membuatnya. Perhatikan bahwa key pair harus memiliki wilayah AWS yang sama dengan instans. Contohnya menggunakan US West (California N.).

Setelah Anda memilih key pair, buat subdirektori `opsworks_cookbooks` bernama `ec2_keys` dan salin file private key (`.pem`) key pair ke subdirektori tersebut. Perhatikan bahwa menempatkan kunci pribadi `ec2_keys` hanyalah kenyamanan yang menyederhanakan kode sedikit; itu bisa di mana saja di sistem Anda.

4. Buat subdirektori `opsworks_cookbooks` bernama `createdir-ec2` dan navigasikan ke sana.
5. Tambahkan `metadata.rb` file `createdir-ec2` dengan konten berikut.

```
name "createdir-ec2"
version "0.1.0"
```

6. Inisialisasi Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket. Bagian berikut menjelaskan cara mengkonfigurasi `.kitchen.yml`, yang secara signifikan lebih rumit untuk EC2 instans Amazon.
7. Tambahkan `recipes` subdirektori kecreatedir-ec2.

Mengonfigurasi .kitchen.ymluntuk Amazon EC2

Anda mengonfigurasi `.kitchen.yml` dengan informasi yang dibutuhkan `kitchen-ec2` driver untuk meluncurkan EC2 instans Amazon yang dikonfigurasi dengan tepat. Berikut ini adalah contoh `.kitchen.yml` file untuk instance Amazon Linux di wilayah AS Barat (California N.).

```
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:
```

Anda dapat menggunakan pengaturan default untuk `suites` bagian `provisioner` dan, tetapi Anda harus mengubah default `driver` dan `platforms` pengaturan. Contoh ini menggunakan daftar pengaturan minimal, dan menerima nilai default untuk sisanya. Untuk daftar lengkap `kitchen-ec2` pengaturan, lihat Kitchen: :Ec2: A Test Kitchen Driver for Amazon. EC2

Contoh menetapkan `driver` atribut berikut. Ini mengasumsikan bahwa Anda telah menetapkan akses pengguna dan kunci rahasia ke variabel lingkungan standar, seperti yang dibahas sebelumnya. Pengemudi menggunakan kunci tersebut secara default. Jika tidak, Anda harus secara eksplisit menentukan kunci dengan menambahkan `aws_access_key_id` dan `aws_secret_access_key` ke `driver` atribut, diatur ke nilai kunci yang sesuai.

name

(Wajib) Atribut ini harus diatur ke`ec2`.

aws_ssh_key_id

(Wajib) Nama key pair Amazon EC2 SSH, yang dinamai `US-East1` dalam contoh ini.

transport.ssh_key

(Wajib) File kunci pribadi (`.pem`) untuk kunci yang Anda tentukan`aws_ssh_key_id`. Untuk contoh ini, file diberi nama `US-East1.pem` dan ada di `../opsworks/ec2_keys` direktori.

region

(Wajib) Wilayah AWS instans. Contohnya menggunakan US West (California N.), yang diwakili oleh`us-west-1`).

availability_zone

(Opsional) Availability Zone instance. Jika Anda menghilangkan pengaturan ini, Test Kitchen menggunakan Availability Zone default untuk wilayah tertentu, yaitu `us-west-1b` untuk US

West (California Utara). Namun, zona default mungkin tidak tersedia untuk akun Anda. Dalam hal ini, Anda harus secara eksplisit menentukan Availability Zone. Seperti yang terjadi, akun yang digunakan untuk menyiapkan contoh tidak mendukung`us-west-1b`, jadi contoh secara eksplisit menentukan. `us-west-1c`

require_chef_omnibus

Ketika diatur `ketrue`, pengaturan ini memastikan bahwa penginstal omnibus digunakan untuk menginstal `chef-client` ke semua instance platform.

security_group_ids

(Opsional) Daftar grup keamanan IDs untuk diterapkan pada instance. Pengaturan ini menerapkan grup `default` keamanan untuk instance. Pastikan bahwa aturan masuknya grup keamanan mengizinkan koneksi SSH masuk, atau Test Kitchen tidak akan dapat berkomunikasi dengan instans. Jika Anda menggunakan grup `default` keamanan, Anda mungkin perlu mengeditnya. Untuk informasi selengkapnya, lihat Grup EC2 Keamanan Amazon.

subnet_id

ID subnet target untuk contoh, jika berlaku.

associate_public_ip

Anda dapat meminta Amazon EC2 mengaitkan alamat IP publik dengan instance jika Anda ingin dapat mengakses instance dari Internet.

antarmuka

Jenis konfigurasi nama host yang Anda gunakan untuk mengakses instance. Nilai yang valid adalah `dns`, `public`, `private`, atau `private_dns`. Jika Anda tidak menentukan nilai untuk atribut ini, `kitchen-ec2` mengatur konfigurasi nama host dalam urutan berikut. Jika Anda menghilangkan atribut ini, jenis konfigurasi tidak disetel.

1. Nama DNS
2. Alamat IP publik
3. Alamat IP privat
4. Nama DNS pribadi

> **Important**
>
> Daripada menggunakan kredensi akun Anda untuk akses dan kunci rahasia, Anda harus membuat pengguna dan memberikan kredensi tersebut ke Test Kitchen. Untuk informasi selengkapnya, lihat Praktik Terbaik untuk Mengelola Kunci Akses AWS.
> Berhati-hatilah untuk tidak menempatkan `.kitchen.yml` di lokasi yang dapat diakses publik, seperti mengunggahnya ke repositori publik GitHub atau Bitbucket. Melakukan hal itu memperlihatkan kredensil Anda dan dapat membahayakan keamanan akun Anda.

`kitchen-ec2`Driver menyediakan dukungan default untuk platform berikut:

- 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
- jendela-2012r2
- jendela-2008r2

Jika Anda ingin menggunakan satu atau lebih dari platform ini, tambahkan nama platform yang sesuai`platforms`. `kitchen-ec2`Driver secara otomatis memilih AMI yang sesuai dan menghasilkan nama pengguna SSH. Anda dapat menggunakan platform lain—contoh ini menggunakan Amazon Linux—tetapi Anda harus secara eksplisit menentukan atribut berikut.
`platforms`

name

Nama platform. Contoh ini menggunakan Amazon Linux, jadi `name` diatur ke`amazon`.

sopir

`driver`Atribut, yang meliputi:

- `image_id`— AMI platform, yang harus milik wilayah yang ditentukan. Contohnya menggunakan`ami-ed8e9284`, Amazon Linux AMI dari wilayah AS Barat (California N.).
- `transport.username`— Nama pengguna SSH yang akan digunakan Test Kitchen untuk berkomunikasi dengan instance.

  Gunakan `ec2-user` untuk Amazon Linux. AMIs Orang lain mungkin memiliki nama pengguna yang berbeda.

Ganti kode `.kitchen.yml` dengan contoh, dan tetapkan nilai yang sesuai ke atribut khusus akun seperti. `aws_access_key_id`

Menjalankan Resep

Contoh ini menggunakan resep dariIterasi.

Untuk menjalankan resep

1. Buat file bernama `default.rb` dengan kode berikut dan simpan ke `recipes` folder buku masak.

```
directory "/srv/www/shared" do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

2. Jalankan `kitchen converge` untuk mengeksekusi resep. Perhatikan bahwa perintah ini akan memakan waktu lebih lama untuk diselesaikan daripada contoh sebelumnya karena waktu yang diperlukan untuk meluncurkan dan menginisialisasi EC2 instance Amazon.
3. Buka EC2konsol Amazon, pilih wilayah AS Barat (California Utara)), dan klik Instans di panel navigasi. Anda akan melihat contoh yang baru dibuat dalam daftar.
4. Jalankan `kitchen login` untuk masuk ke instance, seperti yang telah Anda lakukan untuk instance yang berjalan di VirtualBox. Anda akan melihat direktori yang baru dibuat di bawah`/srv`. Anda juga dapat menggunakan klien SSH favorit Anda untuk terhubung ke instance.

Langkah Berikutnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bab ini memandu Anda melalui dasar-dasar bagaimana menerapkan buku masak Chef, tetapi masih banyak lagi:

- Contoh menunjukkan kepada Anda bagaimana menggunakan beberapa sumber daya yang lebih umum digunakan, tetapi ada banyak lagi.

  Untuk sumber daya yang dicakup, contoh hanya menggunakan beberapa atribut dan tindakan yang tersedia. Untuk referensi lengkap, lihat Tentang Sumber Daya dan Penyedia.
- Contoh hanya menggunakan elemen buku masak inti:`recipes`,, `attributes`files, dan`templates`.

  Buku masak juga dapat mencakup berbagai elemen lain, seperti`libraries`,`definitions`, dan`specs`. Untuk informasi selengkapnya, lihat dokumentasi Chef.
- Contoh menggunakan Test Kitchen hanya sebagai cara mudah untuk memulai instance, menjalankan resep, dan masuk ke instance.

  Test Kitchen terutama merupakan platform pengujian yang dapat Anda gunakan untuk menjalankan berbagai tes pada resep Anda. Jika Anda belum melakukannya, ikuti panduan Test Kitchen lainnya, yang memperkenalkan Anda pada fitur pengujiannya.
- Menerapkan Buku Masak untuk Tumpukan AWS OpsWorksmemberikan beberapa contoh yang lebih maju, dan menunjukkan bagaimana menerapkan buku masak untuk AWS OpsWorks Stacks.

Menerapkan Buku Masak untuk Tumpukan AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Dasar-dasar Buku Masakmemperkenalkan Anda ke buku masak dan resep. Contoh di bagian itu sederhana menurut desain dan akan bekerja pada instance apa pun yang mendukung Chef, termasuk instance AWS OpsWorks Stacks. Untuk menerapkan buku masak yang lebih canggih untuk AWS OpsWorks Stacks, Anda biasanya perlu memanfaatkan sepenuhnya lingkungan AWS OpsWorks Stacks, yang berbeda dari Chef standar dalam beberapa cara.

Topik ini menjelaskan dasar-dasar penerapan resep untuk instance AWS OpsWorks Stacks.

> **Note**
>
> Jika Anda tidak terbiasa dengan cara menerapkan buku masak, Anda harus mulai dengan Dasar-dasar Buku Masak.

Topik

- Menjalankan Resep pada Instans AWS OpsWorks Stacks Linux
- Menjalankan Resep pada Instance Windows
- Menjalankan PowerShell Skrip Windows
- Mengejek Konfigurasi Stack dan Atribut Deployment di Vagrant
- Menggunakan Konfigurasi Stack dan Nilai Atribut Deployment
- Menggunakan Cookbook Eksternal pada Instance Linux: Berkshelf
- Menggunakan SDK for Ruby: Mengunduh File dari Amazon S3
- Menginstal Perangkat Lunak Windows
- Mengesampingkan Atribut Bawaan
- Mengesampingkan Template Bawaan

Menjalankan Resep pada Instans AWS OpsWorks Stacks Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Test Kitchen dan Vagrant menyediakan cara sederhana dan efisien untuk mengimplementasikan buku masak, tetapi untuk memverifikasi bahwa resep buku masak akan berjalan dengan benar dalam produksi, Anda harus menjalankannya pada instance AWS OpsWorks Stacks. Topik ini menjelaskan cara menginstal buku masak khusus pada instance AWS OpsWorks Stacks Linux dan menjalankan resep sederhana. Topik ini juga memberikan beberapa tips untuk memperbaiki bug resep secara efisien.

Untuk deskripsi tentang cara menjalankan resep pada instance Windows, lihatMenjalankan Resep pada Instance Windows.

Topik

- Membuat dan Menjalankan Resep
- Mengeksekusi Resep Secara Otomatis
- Memecahkan Masalah dan Memperbaiki Resep

Membuat dan Menjalankan Resep

Pertama, Anda perlu membuat tumpukan. Berikut ini secara singkat merangkum cara membuat tumpukan untuk contoh ini. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Untuk membuat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan klik Add Stack.
2. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan klik Tambahkan Tumpukan.

   - Nama - OpsTest

- Kunci SSH default - Sebuah EC2 key pair Amazon

Jika Anda perlu membuat EC2 key pair Amazon, lihat Amazon EC2 Key Pairs. Perhatikan bahwa key pair harus memiliki wilayah AWS yang sama dengan instans. Contoh menggunakan wilayah AS Barat (Oregon) default.

3. Klik Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama - OpsTest
   - Nama pendek — opstest

   Tipe lapisan apa pun sebenarnya akan berfungsi untuk tumpukan Linux, tetapi contohnya tidak memerlukan paket apa pun yang diinstal oleh jenis lapisan lain, jadi lapisan khusus adalah pendekatan yang paling sederhana.
4. Tambahkan instance 24/7 dengan pengaturan default ke layer dan mulai.

Saat instance dimulai — biasanya membutuhkan waktu beberapa menit — Anda dapat membuat buku masak. Contoh ini akan menggunakan versi resep yang sedikit dimodifikasiLogika Bersyarat, yang membuat direktori data yang namanya tergantung pada platform.

Untuk mengatur buku masak

1. Buat direktori dalam `opsworks_cookbooks` bernama `opstest` dan navigasikan ke sana.
2. Buat `metadata.rb` file dengan konten berikut dan simpan keo`pstest`.

```
name "opstest"
version "0.1.0"
```

3. Buat `recipes` direktori di dalamnyao`pstest`.
4. Buat `default.rb` file dengan resep berikut dan simpan ke `recipes` direktori.

```
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
```

Perhatikan bahwa resep mencatat pesan, tetapi ia melakukannya dengan meneleponChef::Log.info. Anda tidak menggunakan Test Kitchen untuk contoh ini, jadi log metode ini tidak terlalu berguna. Chef::Log.infomenempatkan pesan ke dalam log Chef, yang dapat Anda baca setelah menjalankan Chef selesai. AWS OpsWorks Stacks menyediakan cara mudah untuk melihat log ini, seperti yang dijelaskan nanti.

> **Note**
>
> Log koki biasanya berisi banyak informasi rutin dan relatif tidak menarik. Karakter '*' yang menghalangi teks pesan membuatnya lebih mudah dikenali.

5. Buat .zip arsipopsworks_cookbooks. Untuk menginstal buku masak Anda pada instance AWS OpsWorks Stacks, Anda harus menyimpannya di repositori dan menyediakan AWS OpsWorks Stacks dengan informasi yang diperlukan untuk mengunduh buku masak ke instance. Anda dapat menyimpan buku masak Anda di salah satu dari beberapa jenis repositori yang didukung. Contoh ini menyimpan file arsip yang berisi buku masak di bucket Amazon S3. Untuk informasi lebih lanjut tentang repositori buku masak, lihat. Repositori

   > **Note**
   >
   > Untuk mempermudah, contoh ini hanya mengarsipkan seluruh opsworks_cookbooks direktori. Namun, itu berarti AWS OpsWorks Stacks akan mengunduh semua buku masak opsworks_cookbooks ke instance, meskipun Anda hanya akan menggunakan salah satunya. Untuk menginstal hanya contoh buku masak, buat direktori induk lain dan pindah opstest ke direktori itu. Kemudian buat .zip arsip direktori induk dan gunakan sebagai gantinyaopsworks_cookbooks.zip.

> Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

6. Unggah arsip ke bucket Amazon S3, buat arsip publik, dan rekam URL arsip.

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya.

Untuk menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus, dan tentukan pengaturan berikut.
   - Jenis repositori - Arsip S3
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

   Gunakan nilai default untuk pengaturan lain dan klik Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Perbarui Buku Masak Kustom, yang menginstal versi buku masak kustom Anda saat ini pada instance tumpukan. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.
3. Jalankan resep dengan menjalankan perintah tumpukan Execute Recipes dengan Recipes untuk mengeksekusi set ke**`opstest::default`**. Perintah ini memulai menjalankan Chef, dengan daftar run yang terdiri dari. `opstest::default`

Setelah resep berjalan dengan sukses, Anda dapat memverifikasinya.

Untuk memverifikasi opstest

1. Langkah pertama adalah memeriksa log Chef. Klik tampilkan di kolom Log instans opstest1 untuk menampilkan log. Gulir ke bawah dan Anda akan melihat pesan log Anda di dekat bagian bawah.

```
...
[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. Gunakan SSH untuk masuk ke instance dan daftar isi. `/srv/www/`

Jika Anda mengikuti semua langkah, Anda akan melihat `/srv/www/config` bukan `/srv/www/shared` direktori yang Anda harapkan. Bagian berikut memberikan beberapa pedoman untuk memperbaiki bug tersebut dengan cepat.

Mengeksekusi Resep Secara Otomatis

Perintah Execute Recipes adalah cara mudah untuk menguji resep khusus, itulah sebabnya ia digunakan di sebagian besar contoh ini. Namun, dalam praktiknya, Anda biasanya menjalankan resep pada titik standar dalam siklus hidup instans, seperti setelah instance selesai booting atau saat Anda menerapkan aplikasi. AWS OpsWorks Stacks menyederhanakan menjalankan resep pada instans Anda dengan mendukung serangkaian peristiwa siklus hidup untuk setiap lapisan: Setup, Configure, Deploy, Undeploy, dan Shutdown. Anda dapat meminta AWS OpsWorks Stacks menjalankan resep secara otomatis pada instance layer dengan menetapkan resep ke peristiwa siklus hidup yang sesuai.

Anda biasanya akan membuat direktori segera setelah instance selesai booting, yang sesuai dengan peristiwa Setup. Berikut ini menunjukkan cara menjalankan contoh resep saat penyiapan, menggunakan tumpukan yang sama yang Anda buat sebelumnya dalam contoh. Anda dapat menggunakan prosedur yang sama untuk acara lainnya.

Untuk menjalankan resep secara otomatis saat penyiapan

1. Pilih Layers di panel navigasi dan kemudian pilih ikon pensil di sebelah link Recipes OpsTest layer.
2. Tambahkan **`opstest::default`** ke resep Setup layer, klik + untuk menambahkannya ke layer, dan pilih Save untuk menyimpan konfigurasi.
3. Pilih Instances, tambahkan instance lain ke layer, dan mulai.

   Instance harus diberi nama`opstest2`. Setelah selesai booting, AWS OpsWorks Stacks akan berjalan. `opstest::default`
4. Setelah `opstest2` instance online, verifikasi yang `/srv/www/shared` ada.

> **Note**
>
> Jika Anda telah menetapkan resep ke acara Setup, Configure, atau Deploy, Anda juga menjalankannya secara manual dengan menggunakan perintah stack (Setup and Configure) atau perintah deploy (Deploy) untuk memicu peristiwa. Perhatikan bahwa jika Anda memiliki beberapa resep yang ditetapkan untuk suatu peristiwa, perintah ini menjalankan semuanya.

Memecahkan Masalah dan Memperbaiki Resep

Jika Anda tidak mendapatkan hasil yang diharapkan, atau resep Anda bahkan tidak berjalan dengan sukses, pemecahan masalah biasanya dimulai dengan memeriksa log Chef. Ini berisi deskripsi rinci tentang proses dan termasuk pesan log inline dari resep Anda. Log sangat berguna jika resep Anda gagal. Ketika itu terjadi, Chef mencatat kesalahan, termasuk jejak tumpukan.

Jika resepnya berhasil, seperti contoh ini, log Chef seringkali tidak banyak membantu. Dalam hal ini, Anda dapat mengetahui masalahnya dengan hanya melihat lebih dekat resepnya, beberapa baris pertama khususnya:

```
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 adalah stand-in yang masuk akal untuk Amazon Linux saat Anda menguji resep di Vagrant, tetapi sekarang Anda menjalankan instance Amazon Linux yang sebenarnya. Nilai platform untuk Amazon Linux adalah`amazon`, yang tidak termasuk dalam `value_for_platform` panggilan, sehingga resep dibuat secara `/srv/www/config` default. Untuk informasi selengkapnya tentang pemecahan masalah, lihat. Panduan Debugging dan Pemecahan Masalah

Sekarang setelah Anda mengidentifikasi masalahnya, Anda perlu memperbarui resep dan memverifikasi perbaikannya. Anda dapat kembali ke file sumber asli, memperbarui`default.rb`, mengunggah arsip baru ke Amazon S3, dan sebagainya. Namun, proses itu bisa sedikit membosankan dan memakan waktu. Berikut ini menunjukkan pendekatan yang jauh lebih cepat yang sangat berguna untuk bug resep sederhana seperti yang ada di contoh: edit resep pada instance.

Untuk mengedit resep pada sebuah instance

1. Gunakan SSH untuk masuk ke instance dan kemudian jalankan `sudo su` untuk meningkatkan hak istimewa Anda. Anda memerlukan hak akses root untuk mengakses direktori buku masak.
2. AWS OpsWorks Stacks menyimpan buku masak Anda`/opt/aws/opsworks/current/site-cookbooks`, jadi navigasikan ke. `/opt/aws/opsworks/current/site-cookbooks/opstest/recipes`

   > **Note**
   >
   > AWS OpsWorks Stacks juga menyimpan salinan buku masak Anda di. `/opt/aws/opsworks/current/merged-cookbooks` Jangan edit buku masak itu. Saat Anda menjalankan resep, AWS OpsWorks Stacks menyalin buku masak dari `.../site-cookbooks` ke`.../merged-cookbooks`, sehingga setiap perubahan yang Anda buat `.../merged-cookbooks` akan ditimpa.

3. Gunakan editor teks pada instance untuk mengeditdefault.rb, dan ganti `centos` denganamazon. Resep Anda sekarang harus terlihat seperti berikut ini.

   ```
   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"
   )
   ...
   ```

Untuk memverifikasi perbaikan, jalankan resep dengan menjalankan perintah stack Execute Recipe lagi. Instance sekarang harus memiliki `/srv/www/shared` direktori. Jika Anda perlu membuat perubahan lebih lanjut pada resep, Anda dapat menjalankan Execute Recipe sesering yang Anda suka; Anda tidak perlu berhenti dan memulai ulang instance setiap kali Anda menjalankan perintah. Ketika Anda puas bahwa resepnya berfungsi dengan benar, jangan lupa untuk memperbarui kode di buku masak sumber Anda.

> **Note**
>
> Jika Anda telah menetapkan resep Anda ke acara siklus hidup sehingga AWS OpsWorks Stacks menjalankannya secara otomatis, Anda selalu dapat menggunakan Execute Recipe untuk menjalankan kembali resep. Anda juga dapat menjalankan kembali resep sebanyak yang Anda inginkan tanpa memulai ulang instance dengan menggunakan konsol AWS OpsWorks Stacks untuk memicu peristiwa yang sesuai secara manual. Namun, pendekatan ini menjalankan semua resep acara. Berikut pengingat:
>
> - Gunakan perintah tumpukan untuk memicu peristiwa Pengaturan atau Konfigurasi.
> - Gunakan perintah deploy untuk memicu peristiwa Deploy atau Undeploy.

Menjalankan Resep pada Instance Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini pada dasarnya adalah versi singkat dariMenjalankan Resep pada Instance Linux, yang menunjukkan kepada Anda cara menjalankan resep di tumpukan Windows. Kami menyarankan Anda untuk melakukannya Menjalankan Resep pada Instance Linux terlebih dahulu, karena ini memberikan diskusi yang lebih rinci, yang sebagian besar relevan dengan kedua jenis sistem operasi.

Untuk penjelasan tentang cara menjalankan resep pada instance AWS OpsWorks Stacks Linux, lihat. Menjalankan Resep pada Instance Linux

Topik

- Mengaktifkan Akses RDP
- Membuat dan Menjalankan Resep
- Mengeksekusi Resep Secara Otomatis

Mengaktifkan Akses RDP

Sebelum memulai, jika Anda belum melakukannya, Anda harus menyiapkan grup keamanan dengan aturan masuk yang memungkinkan akses RDP untuk instans Anda. Anda akan membutuhkan grup itu saat Anda membuat tumpukan.

Saat Anda membuat tumpukan pertama di suatu wilayah, AWS OpsWorks Stacks membuat satu set grup keamanan. Mereka termasuk satu bernama sesuatu seperti`AWS-OpsWorks-RDP-Server`, yang AWS OpsWorks Stacks melampirkan ke semua instance Windows untuk memungkinkan akses RDP. Namun, secara default, grup keamanan ini tidak memiliki aturan apa pun, jadi Anda harus menambahkan aturan masuk untuk mengizinkan akses RDP ke instans Anda.

Untuk memungkinkan akses RDP

1. Buka EC2konsol Amazon, atur ke wilayah tumpukan, dan pilih Grup Keamanan dari panel navigasi.
2. Pilih AWS- OpsWorks -RDP-Server, pilih tab Inbound, dan pilih Edit.
3. Tambahkan aturan dengan pengaturan berikut:
   - Jenis - RDP
   - Sumber — Alamat IP sumber yang diizinkan.

     Anda biasanya mengizinkan permintaan RDP masuk dari alamat IP Anda atau rentang alamat IP tertentu (biasanya rentang alamat IP perusahaan Anda).

> **Note**
>
> Seperti yang dijelaskan nanti, Anda juga harus mengedit izin pengguna untuk mengotorisasi akses RDP untuk pengguna biasa.

Untuk informasi selengkapnya, lihat Masuk dengan RDP.

Membuat dan Menjalankan Resep

Berikut ini secara singkat merangkum cara membuat tumpukan untuk contoh ini. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan pilih Add Stack. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan pilih Tambah Tumpukan.

   - Nama - WindowsRecipeTest
   - Wilayah - AS Barat (Oregon)

     Contoh ini akan bekerja di wilayah mana pun, tetapi kami sarankan menggunakan US West (Oregon) untuk tutorial.
   - Sistem operasi default - Microsoft Windows Server 2012 R2
2. Pilih Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama - RecipeTest
   - Nama pendek — recietest
3. Tambahkan instance 24/7 dengan pengaturan default ke RecipeTest layer dan mulai.

   AWS OpsWorks Stacks secara otomatis menetapkan `AWS-OpsWorks-RDP-Server` ke instance ini, yang memungkinkan pengguna yang berwenang untuk masuk ke instance.
4. Pilih Izin dan kemudian Edit, dan pilih SSH/RDP dan sudo/admin. Pengguna reguler memerlukan otorisasi ini selain grup `AWS-OpsWorks-RDP-Server` keamanan untuk masuk ke instance.

   > **Note**
   >
   > Anda juga dapat masuk sebagai Administrator, tetapi memerlukan prosedur yang berbeda. Untuk informasi selengkapnya, lihat Masuk dengan RDP.

Saat instance dimulai — biasanya membutuhkan waktu beberapa menit — Anda dapat membuat buku masak. Resep untuk contoh ini membuat direktori data, dan pada dasarnya resep dariContoh 3: Membuat Direktori, dimodifikasi untuk Windows.

> **Note**
>
> Saat menerapkan buku masak untuk instance AWS OpsWorks Stacks Windows, Anda menggunakan struktur direktori yang agak berbeda dari yang Anda lakukan saat

mengimplementasikan buku masak untuk AWS OpsWorks instance Stacks Linux. Untuk informasi selengkapnya, lihat Repositori.

Untuk mengatur buku masak

1. Buat direktori bernama `windowstest` dan navigasikan ke sana.
2. Buat `metadata.rb` file dengan konten berikut dan simpan ke`windowstest`.

```
name "windowstest"
version "0.1.0"
```

3. Buat `recipes` direktori di dalamnya`windowstest`.
4. Buat `default.rb` file dengan resep berikut dan simpan ke `recipes` direktori.

```
Chef::Log.info("******Creating a data directory.******")

directory 'C:\data' do
  rights :full_control, 'instance_name\username'
  inherits false
  action :create
end
```

   Ganti *`username`* dengan nama pengguna Anda.

5. Masukkan buku masak ke dalam repositori.

   Untuk menginstal buku masak Anda pada instance AWS OpsWorks Stacks, Anda harus menyimpannya di repositori dan menyediakan AWS OpsWorks Stacks dengan informasi yang diperlukan untuk mengunduh buku masak ke instance. Anda dapat menyimpan buku masak Windows sebagai file arsip di bucket S3 atau di repositori Git. Contoh ini menggunakan bucket S3, jadi Anda harus membuat arsip.zip direktori. `windowstest` Untuk informasi lebih lanjut tentang repositori buku masak, lihat. Repositori

6. Unggah arsip ke bucket S3, buat arsip publik, dan rekam URL arsip. Anda juga dapat menggunakan arsip pribadi, tetapi arsip publik sudah cukup untuk contoh ini dan agak lebih mudah untuk dikerjakan.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya.

Untuk menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip S3
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

   Terima nilai default untuk pengaturan lain dan pilih Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Perbarui Buku Masak Kustom, yang menginstal versi buku masak kustom Anda saat ini pada instance tumpukan, termasuk instance online. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.
3. Setelah Update Custom Cookbooks selesai, jalankan resep dengan menjalankan perintah stack Execute Recipes dengan Recipes untuk mengeksekusi set ke**`windowstest::default`**. Perintah ini memulai menjalankan Chef, dengan daftar run yang terdiri dari resep Anda.

Setelah resep berjalan dengan sukses, Anda dapat memverifikasinya.

Untuk memverifikasi windowstest

1. Periksa log Chef. Pilih tampilkan di kolom Log instans opstest1 untuk menampilkan log. Gulir ke bawah dan Anda akan melihat pesan log Anda di dekat bagian bawah.

```
...
[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. Pilih Instans, pilih rdp di kolom Tindakan instans, dan minta kata sandi RDP dengan waktu kedaluwarsa yang sesuai. Salin nama DNS, nama pengguna, dan kata sandi. Anda kemudian dapat menggunakan informasi tersebut dengan klien RDP, seperti klien Windows Remote Desktop Connection, untuk masuk ke instance dan memverifikasi yang ada `c:\data`. Untuk informasi selengkapnya, lihat Masuk dengan RDP.

> **Note**
>
> Jika resep Anda tidak berfungsi dengan baik, lihat Memecahkan Masalah dan Memperbaiki Resep untuk tips pemecahan masalah; kebanyakan dari mereka juga berlaku untuk instance Windows. Jika Anda ingin menguji perbaikan Anda dengan mengedit resep pada instance, cari buku masak Anda di `C:\chef\cookbooks` direktori, tempat AWS OpsWorks Stacks menginstal buku masak khusus.

Mengeksekusi Resep Secara Otomatis

Perintah Execute Recipes adalah cara mudah untuk menguji resep khusus, itulah sebabnya ia digunakan di sebagian besar contoh ini. Namun, dalam praktiknya, Anda biasanya menjalankan resep pada titik standar dalam siklus hidup instans, seperti setelah instance selesai booting atau saat Anda menerapkan aplikasi. AWS OpsWorks Stacks menyederhanakan menjalankan resep pada instans Anda dengan mendukung serangkaian peristiwa siklus hidup untuk setiap lapisan: Setup, Configure, Deploy, Undeploy, dan Shutdown. Anda dapat meminta AWS OpsWorks Stacks menjalankan resep secara otomatis pada instance layer dengan menetapkan resep ke peristiwa siklus hidup yang sesuai.

Anda biasanya akan membuat direktori segera setelah instance selesai booting, yang sesuai dengan peristiwa Setup. Berikut ini menunjukkan cara menjalankan contoh resep saat penyiapan, menggunakan tumpukan yang sama yang Anda buat sebelumnya dalam contoh. Anda dapat menggunakan prosedur yang sama untuk acara lainnya.

Untuk menjalankan resep secara otomatis saat penyiapan

1. Pilih Layers di panel navigasi dan kemudian pilih ikon pensil di sebelah link Recipes RecipeTest layer.

2. Tambahkan **`windowstest::default`** ke resep Setup layer, pilih + untuk menambahkannya ke layer, dan pilih Save untuk menyimpan konfigurasi.
3. Pilih Instances, tambahkan instance lain ke layer, dan mulai.

   Instance harus diberi nama`recipetest2`. Setelah selesai booting, AWS OpsWorks Stacks akan berjalan. `windowstest::default`
4. Setelah `recipetest2` instance online, verifikasi yang `c:\data` ada.

> **Note**
>
> Jika Anda telah menetapkan resep ke acara Setup, Configure, atau Deploy, Anda juga dapat menjalankannya secara manual dengan menggunakan perintah stack (Setup and Configure) atau perintah deploy (Deploy) untuk memicu peristiwa. Perhatikan bahwa jika Anda memiliki beberapa resep yang ditetapkan untuk suatu peristiwa, perintah ini menjalankan semuanya.

Menjalankan PowerShell Skrip Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Contoh-contoh ini mengasumsikan bahwa Anda telah melakukan Menjalankan Resep pada Instance Windows contoh. Jika tidak, Anda harus melakukan contoh itu terlebih dahulu. Secara khusus, ini menjelaskan cara mengaktifkan akses RDP ke instance Anda.

Salah satu cara agar resep melakukan tugas pada instance Windows — terutama tugas yang tidak memiliki sumber daya Chef yang sesuai — adalah dengan meminta resep menjalankan skrip

Windows. PowerShell Bagian ini memperkenalkan Anda pada dasar-dasar dengan menjelaskan cara menggunakan PowerShell skrip Windows untuk menginstal fitur Windows.

Sumber `powershell_script`daya menjalankan PowerShell cmdlet Windows pada sebuah instance. Contoh berikut menggunakan Install- WindowsFeature cmdlet untuk menginstal penampil XPS pada instance.

Berikut ini secara singkat merangkum cara membuat tumpukan untuk contoh ini. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan pilih Add Stack. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan klik Tambahkan Tumpukan.

   - Nama - PowerShellTest
   - Wilayah - AS Barat (Oregon)

     Contoh ini akan bekerja di wilayah mana pun, tetapi kami sarankan menggunakan US West (Oregon) untuk tutorial.
   - Sistem operasi default - Microsoft Windows Server 2012 R2
2. Pilih Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama - PowerShell
   - Nama pendek - PowerShell
3. Tambahkan instance 24/7 dengan pengaturan default ke PowerShell layer dan mulailah.
4. Pilih Izin dan kemudian Edit, dan pilih SSH/RDP dan sudo/admin. Anda memerlukan otorisasi ini selain grup `AWS-OpsWorks-RDP-Server` keamanan untuk masuk ke instance sebagai pengguna biasa.

Saat instance dimulai — biasanya membutuhkan waktu beberapa menit — Anda dapat membuat buku masak. Resep untuk contoh ini membuat direktori data, dan pada dasarnya resep dariContoh 3: Membuat Direktori, dimodifikasi untuk Windows.

Untuk mengatur buku masak

1. Buat direktori bernama `powershell` dan navigasikan ke sana.

2. Buat `metadata.rb` file dengan konten berikut dan simpan kewindowstest.

```
name "powershell"
version "0.1.0"
```

3. Buat `recipes` direktori di dalam `powershell` direktori.
4. Buat `default.rb` file dengan resep berikut dan simpan ke `recipes` direktori.

```
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
```

   - Sumber `powershell_script` daya menjalankan cmdlet untuk menginstal penampil XPS.

     Contoh ini hanya menjalankan satu cmdlet, tetapi `code` blok dapat berisi sejumlah baris perintah.
   - `guard_interpreter`Atribut mengarahkan Chef untuk menggunakan Windows PowerShell versi 64-bit.
   - Atribut `not_if` guard memastikan bahwa Chef tidak menginstal fitur jika sudah diinstal.
5. Buat `.zip` arsip `powershell` direktori.
6. Unggah arsip ke bucket Amazon S3, buat arsip publik, dan rekam URL arsip. Anda juga dapat menggunakan arsip pribadi, tetapi arsip publik sudah cukup untuk contoh ini, dan agak lebih mudah untuk dikerjakan.

   Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya.

Untuk menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip S3
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

   Terima nilai default untuk pengaturan lain dan pilih Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Update Custom Cookbooks untuk menginstal versi buku masak kustom Anda saat ini pada instance.
3. Setelah Update Custom Cookbooks selesai, jalankan resep dengan menjalankan perintah stack Execute Recipes dengan Recipes untuk mengeksekusi set ke**`powershell::default`**.

> **Note**
>
> Contoh ini menggunakan Execute Recipes untuk kenyamanan, tetapi Anda biasanya memiliki AWS OpsWorks Stacks menjalankan resep Anda secara otomatis dengan menetapkannya ke acara siklus hidup yang sesuai. Anda dapat menjalankan resep tersebut dengan memicu acara secara manual. Anda dapat menggunakan perintah stack untuk memicu peristiwa Setup dan Configure, dan perintah deploy untuk memicu peristiwa Deploy dan Undeploy.

Setelah resep berjalan dengan sukses, Anda dapat memverifikasinya.

Untuk memverifikasi resep PowerShell

1. Periksa log Chef. Klik tampilkan di kolom Log instans PowerShell1 untuk menampilkan log. Gulir ke bawah dan Anda akan melihat pesan log Anda di dekat bagian bawah.

```
...
[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. Gunakan RDP untuk masuk ke instance dan buka menu Start. XPS Viewer harus terdaftar dengan Aksesori Windows.

Mengejek Konfigurasi Stack dan Atribut Deployment di Vagrant

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Topik ini hanya berlaku untuk instance Linux. Test Kitchen belum mendukung Windows, jadi Anda akan menjalankan semua contoh Windows pada instance AWS OpsWorks Stacks.

AWS OpsWorks Stacks menambahkan konfigurasi tumpukan dan atribut penerapan ke objek node untuk setiap instance di tumpukan Anda untuk setiap peristiwa siklus hidup. Atribut ini menyediakan snapshot konfigurasi tumpukan, termasuk konfigurasi setiap lapisan dan instance online-nya, konfigurasi setiap aplikasi yang diterapkan, dan sebagainya. Karena atribut ini berada di objek node, mereka dapat diakses oleh resep apa pun; sebagian besar resep untuk instance AWS OpsWorks Stacks menggunakan satu atau lebih atribut ini.

Instance yang berjalan di kotak Vagrant tidak dikelola oleh AWS OpsWorks Stacks, sehingga objek simpulnya tidak menyertakan konfigurasi tumpukan dan atribut penerapan apa pun secara default. Namun, Anda dapat menambahkan serangkaian atribut yang sesuai ke lingkungan Test Kitchen. Test Kitchen kemudian menambahkan atribut ke objek node instance, dan resep Anda dapat mengakses atribut seperti pada instance AWS OpsWorks Stacks.

Topik ini menunjukkan cara mendapatkan salinan konfigurasi tumpukan dan atribut penerapan yang sesuai, menginstal atribut pada instance, dan mengaksesnya.

> **Note**
>
> Jika Anda menggunakan Test Kitchen untuk menjalankan pengujian pada resep Anda, fauxhai menyediakan cara alternatif untuk mengejek konfigurasi tumpukan dan penerapan JSON.

Untuk mengatur buku masak

1. Buat subdirektori `opsworks_cookbooks` bernama `printjson` dan navigasikan ke sana.
2. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket.
3. Tambahkan dua subdirektori ke`printjson`: `recipes` dan. `environments`

Anda dapat mengejek konfigurasi tumpukan dan atribut penerapan dengan menambahkan file atribut ke buku masak Anda dengan definisi yang sesuai, tetapi pendekatan yang lebih baik adalah menggunakan lingkungan Test Kitchen. Ada dua pendekatan dasar:

- Tambahkan definisi atribut ke`.kitchen.yml`.

  Pendekatan ini paling berguna jika Anda hanya memiliki beberapa atribut. Untuk informasi lebih lanjut, lihat kitchen.yl.
- Tentukan atribut dalam file lingkungan dan referensi file di`.kitchen.yml`.

  Pendekatan ini biasanya lebih disukai untuk konfigurasi tumpukan dan atribut penyebaran karena file lingkungan sudah dalam format JSON. Anda bisa mendapatkan salinan atribut dalam format JSON dari instance AWS OpsWorks Stacks yang sesuai dan hanya menempelkannya. Semua contoh menggunakan file lingkungan.

Cara termudah untuk membuat konfigurasi tumpukan dan atribut penerapan untuk buku masak Anda adalah dengan membuat tumpukan yang dikonfigurasi dengan tepat dan menyalin atribut yang dihasilkan dari sebuah instance sebagai JSON. Agar file lingkungan Test Kitchen Anda tetap dapat dikelola, Anda kemudian dapat mengedit JSON itu agar hanya memiliki atribut yang dibutuhkan resep Anda. Contoh dalam Bab ini didasarkan pada tumpukan dariMemulai dengan Chef 11 Linux Stacks,

yang merupakan tumpukan server aplikasi PHP sederhana dengan penyeimbang beban, server aplikasi PHP, dan server database MySQL.

Untuk membuat konfigurasi tumpukan dan penerapan JSON

1. Buat MyStack seperti yang dijelaskan dalamMemulai dengan Chef 11 Linux Stacks, termasuk menerapkan SimplePHPApp. Jika mau, Anda dapat menghilangkan instance PHP App Server kedua yang dipanggilLangkah 4: Scale Out MyStack; contoh tidak menggunakan atribut tersebut.
2. Jika Anda belum melakukannya, mulai `php-app1` instance, lalu masuk dengan SSH.
3. Di jendela terminal, jalankan perintah agent cli berikut:

   ```
   sudo opsworks-agent-cli get_json
   ```

   Perintah ini mencetak konfigurasi tumpukan terbaru dan atribut penerapan instance ke jendela terminal dalam format JSON.
4. Salin JSON ke `.json` file dan simpan di lokasi yang nyaman di workstation Anda. Detailnya tergantung pada klien SSH Anda. Misalnya, jika Anda menggunakan PuTTY di Windows, Anda dapat menjalankan `Copy All to Clipboard` perintah, yang menyalin semua teks di jendela terminal ke clipboard Windows. Anda kemudian dapat menempelkan konten ke dalam `.json` file dan mengedit file untuk menghapus teks asing.
5. Edit MyStack JSON sesuai kebutuhan. Konfigurasi tumpukan dan atribut penerapan sangat banyak, dan buku masak biasanya hanya menggunakan sebagian kecil saja. Agar file lingkungan Anda tetap terkelola, Anda dapat mengedit JSON sehingga mempertahankan struktur aslinya tetapi hanya berisi atribut yang benar-benar digunakan oleh buku masak Anda.

   Contoh ini menggunakan versi MyStack JSON yang banyak diedit yang mencakup hanya dua `['opsworks']['stack']` atribut, `['id]` dan. `['name']` Buat versi MyStack JSON yang diedit yang terlihat seperti berikut ini:

   ```
   {
     "opsworks": {
       "stack": {
         "name": "MyStack",
         "id": "42dfd151-6766-4f1c-9940-ba79e5220b58",
       },
     },
   }
   ```

Untuk mendapatkan JSON ini ke objek node instance, Anda perlu menambahkannya ke lingkungan Test Kitchen.

Untuk menambahkan konfigurasi tumpukan dan atribut penerapan ke lingkungan Test Kitchen

1. Buat file lingkungan bernama `test.json` dengan konten berikut dan simpan ke `environments` folder buku masak.

```
{
  "default_attributes": {
    "opsworks" : {
      "stack" : {
        "name" : "MyStack",
        "id" : "42dfd151-6766-4f1c-9940-ba79e5220b58"
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

   File lingkungan memiliki elemen-elemen berikut:

   - `default_attributes`— Atribut default dalam format JSON.

     Atribut ini ditambahkan ke objek node dengan tipe `default` atribut, yang merupakan tipe yang digunakan oleh semua konfigurasi tumpukan dan penerapan atribut JSON. Contoh ini menggunakan versi yang diedit dari konfigurasi tumpukan dan penerapan JSON yang ditunjukkan sebelumnya.
   - `chef_type`— Atur elemen ini ke`environment`.
   - `json_class`— Atur elemen ini ke`Chef::Environment`.
2. Edit `.kitchen.yml` untuk menentukan lingkungan Test Kitchen, sebagai berikut.

```
---
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:
```

Anda menentukan lingkungan dengan menambahkan elemen berikut ke default yang `.kitchen.yml` dibuat oleh`kitchen init`.

penyedia

Tambahkan elemen-elemen berikut.

- `name`— Atur elemen ini ke`chef_solo`.

  Untuk mereplikasi lingkungan AWS OpsWorks Stacks lebih dekat, Anda dapat menggunakan mode lokal klien Chef alih-alih Chef solo. Mode lokal adalah opsi klien Chef yang menggunakan versi ringan dari server Chef (Chef Zero) yang berjalan secara lokal pada instance alih-alih server jarak jauh. Ini memungkinkan resep Anda untuk menggunakan fitur server Chef seperti pencarian atau tas data tanpa terhubung ke server jarak jauh.

- `environments_path`— Subdirektori buku masak yang berisi file lingkungan, `./environments` untuk contoh ini.

setelan:penyedia

Tambahkan `solo_rb` elemen dengan `environment` elemen yang disetel ke nama file lingkungan, dikurangi ekstensi.json. Contoh ini ditetapkan `environment` ke`test`.

3. Buat file resep bernama `default.rb` dengan konten berikut dan simpan ke `recipes` direktori buku masak.

```
log "Stack name: #{node['opsworks']['stack']['name']}"
log "Stack id: #{node['opsworks']['stack']['id']}"
```

Resep ini hanya mencatat dua konfigurasi tumpukan dan nilai penerapan yang Anda tambahkan ke lingkungan. Meskipun resep berjalan secara lokal di Virtual Box, Anda mereferensikan atribut tersebut menggunakan sintaks node yang sama dengan yang Anda lakukan jika resep berjalan pada instance AWS OpsWorks Stacks.

4. Jalankan `kitchen converge`. Anda akan melihat sesuatu seperti output log berikut.

```
...
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

...
```

Menggunakan Konfigurasi Stack dan Nilai Atribut Deployment

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Resep sering membutuhkan informasi tentang konfigurasi tumpukan atau aplikasi yang diterapkan. Misalnya, Anda mungkin memerlukan daftar alamat IP stack untuk membuat file konfigurasi, atau

direktori penerapan aplikasi untuk membuat direktori log. Alih-alih menyimpan data ini di server pusat, AWS OpsWorks Stacks menginstal satu set konfigurasi tumpukan dan atribut penerapan di setiap objek node instance untuk setiap peristiwa siklus hidup. Atribut ini mewakili status tumpukan saat ini, termasuk aplikasi yang diterapkan. Resep kemudian dapat memperoleh data yang mereka butuhkan dari objek node.

> **Note**
>
> Aplikasi terkadang membutuhkan informasi dari objek node, seperti konfigurasi tumpukan dan nilai atribut deployment. Namun, aplikasi tidak dapat mengakses objek node. Untuk memberikan data objek node ke aplikasi, Anda dapat menerapkan resep yang mengambil informasi yang diperlukan dari objek node dan meletakkannya dalam file dalam format yang nyaman. Aplikasi kemudian dapat membaca data dari file. Untuk informasi lebih lanjut dan contoh, lihat Melewati Data ke Aplikasi.

Resep dapat memperoleh konfigurasi stack dan nilai atribut deployment dari objek node sebagai berikut.

- Secara langsung, dengan menggunakan nama atribut yang sepenuhnya memenuhi syarat.

  Anda dapat menggunakan pendekatan ini dengan tumpukan Linux apa pun, tetapi tidak dengan tumpukan Windows.
- Dengan pencarian Chef, yang dapat Anda gunakan untuk menanyakan objek node untuk nilai atribut.

  Anda dapat menggunakan pendekatan ini dengan tumpukan Windows dan tumpukan Linux Chef 11.10.

> **Note**
>
> Dengan tumpukan Linux, Anda dapat menggunakan agen CLI untuk mendapatkan salinan konfigurasi tumpukan instans dan atribut penerapan dalam format JSON. Untuk informasi selengkapnya, lihat Mengejek Konfigurasi Stack dan Atribut Deployment di Vagrant.

Topik

- Memperoleh Nilai Atribut Secara Langsung
- Memperoleh Nilai Atribut dengan Pencarian Chef

Memperoleh Nilai Atribut Secara Langsung

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Pendekatan ini hanya berfungsi untuk tumpukan Linux.

Mengejek Konfigurasi Stack dan Atribut Deployment di Vagrantmenunjukkan cara mendapatkan konfigurasi tumpukan dan data penyebaran dengan menggunakan sintaks node untuk secara langsung mereferensikan atribut tertentu. Ini terkadang merupakan pendekatan terbaik. Namun, banyak atribut didefinisikan dalam koleksi atau daftar yang isi dan namanya dapat bervariasi dari tumpukan ke tumpukan dan dari waktu ke waktu untuk tumpukan tertentu. Misalnya, `deploy` atribut berisi daftar atribut aplikasi, yang diberi nama dengan nama pendek aplikasi. Daftar ini, termasuk nama atribut aplikasi, biasanya bervariasi dari tumpukan ke tumpukan dan bahkan dari penerapan ke penerapan.

Seringkali lebih berguna, dan kadang-kadang bahkan perlu, untuk mendapatkan data yang diperlukan dengan menghitung atribut dalam daftar atau koleksi. Misalnya, misalkan Anda ingin mengetahui alamat IP publik dari instance tumpukan Anda. Informasi itu ada di `['opsworks']['layers']` atribut, yang diatur ke tabel hash yang berisi satu elemen untuk setiap lapisan tumpukan, dinamai dengan nama pendek layer. Setiap elemen layer diatur ke tabel hash yang berisi atribut layer, salah satunya adalah`['instances']`. Elemen itu pada gilirannya diatur ke tabel hash lain yang berisi atribut untuk setiap instance layer, dinamai dengan shortname instance. Setiap atribut instance diatur ke tabel hash lain yang berisi atribut instance, termasuk`['ip']`, yang

mewakili alamat IP publik. Jika Anda mengalami kesulitan memvisualisasikan ini, prosedur berikut menyertakan contoh dalam format JSON.

Contoh ini menunjukkan cara mendapatkan data dari konfigurasi tumpukan dan penyebaran JSON untuk lapisan tumpukan.

Untuk mengatur buku masak

1. Buat direktori dalam `opsworks_cookbooks` bernama `listip` dan navigasikan ke sana.
2. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket.
3. Tambahkan dua direktori ke`listip`: `recipes` dan`environments`.
4. Buat versi JSON yang diedit dari MyStack konfigurasi dan atribut penyebaran yang berisi atribut yang relevan. Seharusnya terlihat seperti berikut ini.

```
{
  "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. Buat file lingkungan bernama`test.json`, tempel contoh JSON ke dalam`default_attributes`, dan simpan file ke folder buku `environments` masak. File akan terlihat seperti berikut (untuk singkatnya, sebagian besar contoh JSON diwakili oleh elipsis).

```
{
  "default_attributes" : {
    "opsworks": {
      "layers": {
        ...
      }
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

6. Ganti teks `.kitchen.yml` dengan yang berikut ini.

```
---
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:
```

Setelah buku masak disiapkan, Anda dapat menggunakan resep berikut untuk mencatat lapisan IDs.

```
node['opsworks']['layers'].each do |layer, layerdata|
  log "#{layerdata['name']} : #{layerdata['id']}"
end
```

Resep menghitung lapisan dalam `['opsworks']['layers']` dan mencatat nama dan ID setiap lapisan.

Untuk menjalankan resep logging ID layer

1. Buat file bernama `default.rb` dengan resep contoh dan simpan ke `recipes` direktori.
2. Jalankan `kitchen converge`.

Bagian yang relevan dari output akan terlihat seperti berikut ini.

```
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
```

Untuk membuat daftar alamat IP instance, Anda memerlukan loop bersarang seperti berikut ini.

```
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
```

Loop bagian dalam mengulangi instance setiap lapisan dan mencatat alamat IP.

Untuk menjalankan contoh resep pencatatan IP

1. Ganti kode `default.rb` dengan resep contoh.
2. Jalankan `kitchen converge` untuk mengeksekusi resep.

Bagian yang relevan dari output akan terlihat seperti berikut ini.

```
  * 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
```

Setelah selesai, jalankan`kitchen destroy`; topik berikutnya menggunakan buku masak baru.

> **Note**
>
> Salah satu alasan paling umum untuk menghitung konfigurasi tumpukan dan pengumpulan JSON penyebaran adalah untuk mendapatkan data untuk aplikasi yang diterapkan tertentu, seperti direktori penerapannya. Sebagai contoh, lihat Menyebarkan Resep.

Memperoleh Nilai Atribut dengan Pencarian Chef

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Pendekatan ini tersedia untuk tumpukan Windows dan tumpukan Linux Chef 11.10.

Mendapatkan konfigurasi tumpukan dan nilai atribut penerapan langsung dari objek node bisa rumit, dan tidak dapat digunakan dengan tumpukan Windows. Pendekatan alternatif adalah dengan menggunakan pencarian Chef untuk menanyakan atribut yang diminati. Jika Anda terbiasa dengan server Chef, Anda akan menemukan bahwa pencarian Chef bekerja sedikit berbeda dengan AWS OpsWorks Stacks. Karena AWS OpsWorks Stacks menggunakan chef-client dalam mode lokal, pencarian Chef bergantung pada versi lokal server Chef yang disebut chef-zero, sehingga pencarian beroperasi pada data yang disimpan secara lokal di objek node instance alih-alih di server jarak jauh.

Sebagai masalah praktis, membatasi pencarian ke data yang disimpan secara lokal biasanya tidak masalah karena objek node pada instance AWS OpsWorks Stacks menyertakan konfigurasi tumpukan dan atribut penerapan. Mereka berisi sebagian besar jika tidak semua data yang biasanya diperoleh resep dari server Chef dan menggunakan nama yang sama, sehingga Anda biasanya dapat menggunakan kode pencarian yang ditulis untuk server Chef pada instance AWS OpsWorks Stacks tanpa modifikasi. Untuk informasi selengkapnya, lihat Menggunakan Pencarian Chef.

Berikut ini menunjukkan struktur dasar dari kueri penelusuran:

```
result = search(:search_index, "key:pattern")
```

- Indeks pencarian menentukan atribut apa yang diterapkan kueri dan menentukan jenis objek yang dikembalikan.
- Kunci menentukan nama atribut.
- Pola menentukan nilai atribut yang ingin Anda ambil.

  Anda dapat menanyakan nilai atribut tertentu atau menggunakan kartu liar untuk menanyakan berbagai nilai.
- Hasilnya adalah daftar objek yang memenuhi kueri, yang masing-masing adalah tabel hash yang berisi beberapa atribut terkait.

  Misalnya, jika Anda menggunakan indeks `node` pencarian, kueri mengembalikan daftar objek instance, satu untuk setiap instance yang memenuhi kueri. Setiap objek adalah tabel hash yang berisi sekumpulan atribut yang menentukan konfigurasi instance, seperti nama host dan alamat IP.

Misalnya, kueri berikut menggunakan indeks `node` pencarian, yang merupakan indeks Chef standar yang berlaku untuk instance tumpukan (atau node, dalam terminologi Chef). Ini mencari contoh dengan nama host dari. `myhost`

```
result = search(:node, "hostname:myhost")
```

Pencarian mengembalikan daftar objek contoh yang hostname adalah`myhost`. Jika Anda menginginkan sistem operasi instance pertama, misalnya, itu akan diwakili oleh`result[0][:os]`. Jika kueri mengembalikan beberapa objek, Anda dapat menghitung mereka untuk mengambil informasi yang diperlukan.

Rincian cara menggunakan pencarian dalam resep tergantung pada apakah Anda menggunakan tumpukan Linux atau Windows. Topik berikut memberikan contoh untuk kedua jenis tumpukan.

Topik

- Menggunakan Pencarian di Linux Stack
- Menggunakan Pencarian di Stack Windows

Menggunakan Pencarian di Linux Stack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Contoh ini didasarkan pada tumpukan Linux dengan satu server aplikasi PHP. Ini menggunakan pencarian Chef untuk mendapatkan alamat IP publik server dan menempatkan alamat dalam file di `/tmp` direktori. Ini mengambil pada dasarnya informasi yang sama dari objek node sepertiMemperoleh Nilai Atribut Secara Langsung , tetapi kode jauh lebih sederhana dan tidak tergantung pada rincian konfigurasi stack dan struktur atribut deployment.

Berikut ini secara singkat merangkum cara membuat tumpukan untuk contoh ini. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

> **Note**
>
> Jika Anda belum menjalankan resep khusus pada instance AWS OpsWorks Stacks sebelumnya, Anda harus terlebih dahulu melihat Menjalankan Resep pada Instance Linux contoh tersebut.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan klik Add Stack.
2. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan klik Tambahkan Tumpukan.

   - Nama — SearchJSON
   - Kunci SSH default - Sebuah EC2 key pair Amazon

   Jika Anda perlu membuat EC2 key pair Amazon, lihat Amazon EC2 Key Pairs. Perhatikan bahwa key pair harus memiliki wilayah AWS yang sama dengan instans. Contohnya menggunakan wilayah AS Barat (Oregon).
3. Klik Add a layer dan tambahkan layer PHP App Server ke stack dengan pengaturan default.
4. Tambahkan instance 24/7 dengan pengaturan default ke layer dan mulai.

Untuk mengatur buku masak

1. Buat direktori dalam `opsworks_cookbooks` bernama `searchjson` dan navigasikan ke sana.
2. Buat `metadata.rb` file dengan konten berikut dan simpan keo`pstest`.

```
name "searchjson"
version "0.1.0"
```

3. Buat `recipes` direktori di dalamnya`searchjson`.
4. Buat `default.rb` file dengan resep berikut dan simpan ke `recipes` direktori.

```
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
```

Tumpukan Linux hanya mendukung indeks node pencarian. Resep menggunakan indeks ini untuk mendapatkan daftar contoh di php-app lapisan. Karena lapisan diketahui hanya memiliki satu contoh, resep hanya menetapkan yang pertama. phpserver Jika layer memiliki beberapa instance, Anda dapat menghitung mereka untuk mengambil informasi yang diperlukan. Setiap item daftar adalah tabel hash yang berisi satu set atribut instance. ipAtribut diatur ke alamat IP publik instans, sehingga Anda dapat mewakili alamat tersebut dalam kode resep berikutnya sebagaiphpserver[:ip].

Setelah menambahkan pesan ke log Chef, resep kemudian menggunakan file sumber daya untuk membuat file bernamaip_addresses. contentAtribut diatur ke representasi string dariphpserver[:ip]. Saat Chef membuatip_addresses, ia menambahkan string itu ke file.

5. Buat .zip arsipopsworks_cookbooks, Unggah arsip ke bucket Amazon S3, buat arsip publik, dan rekam URL arsip. Untuk informasi lebih lanjut tentang repositori buku masak, lihat. Repositori

   Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya.

Untuk menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus, dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip Http
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

   Gunakan nilai default untuk pengaturan lain dan klik Simpan untuk memperbarui konfigurasi tumpukan.

2. Edit konfigurasi lapisan kustom dan tetapkan `searchjson::default` ke acara Setup layer. AWS OpsWorks Tumpukan akan menjalankan resep setelah instance boot atau jika Anda secara eksplisit memicu peristiwa Pengaturan.
3. Jalankan perintah tumpukan Perbarui Buku Masak Kustom, yang menginstal versi repositori buku masak kustom Anda saat ini pada instance tumpukan. Jika versi sebelumnya dari repositori hadir, perintah ini menimpa itu.
4. Jalankan resep dengan menjalankan perintah Setup stack, yang memicu peristiwa Setup pada instance dan berjalan`searchjson::default`. Biarkan halaman pengaturan perintah Running terbuka.

Setelah resep berhasil berjalan, Anda dapat memverifikasinya.

Untuk memverifikasi searchjson

1. Langkah pertama adalah memeriksa log Chef untuk acara Pengaturan terbaru. Pada halaman pengaturan perintah Running, klik tampilkan di kolom Log instans php-app1 untuk menampilkan log. Gulir ke bawah untuk menemukan pesan log Anda di dekat tengah, yang akan terlihat seperti berikut ini.

```
...
[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. Gunakan SSH untuk masuk ke instance dan daftar isi`/tmp`, yang harus menyertakan file bernama `ip_addresses` yang berisi alamat IP.

Menggunakan Pencarian di Stack Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks menyediakan dua opsi untuk menggunakan pencarian di tumpukan Windows.

- Indeks node pencarian, yang dapat digunakan untuk menanyakan satu set atribut Chef standar.

  Jika Anda memiliki resep yang ada dengan kode pencarian yang menggunakannode, mereka biasanya akan bekerja pada AWS OpsWorks tumpukan Stacks tanpa modifikasi.
- Kumpulan indeks penelusuran tambahan yang dapat digunakan untuk menanyakan kumpulan atribut AWS OpsWorks khusus Stacks, dan beberapa atribut standar.

  Indeks ini dibahas dalamMenggunakan Indeks Pencarian AWS OpsWorks Khusus Tumpukan di Tumpukan Windows.

Kami merekomendasikan penggunaan node untuk mengambil informasi standar, seperti nama host atau alamat IP. Pendekatan itu akan membuat resep Anda konsisten dengan praktik Chef standar. Gunakan indeks pencarian AWS OpsWorks Stacks untuk mengambil informasi yang khusus untuk Stacks. AWS OpsWorks

Topik

- Menggunakan indeks pencarian node pada tumpukan Windows
- Menggunakan Indeks Pencarian AWS OpsWorks Khusus Tumpukan di Tumpukan Windows

Menggunakan indeks pencarian node pada tumpukan Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Contoh ini mengasumsikan bahwa Anda telah melakukan Menjalankan Resep pada Instance Windows contoh. Jika tidak, Anda harus melakukan contoh itu terlebih dahulu. Secara khusus, ini menjelaskan cara mengaktifkan akses RDP ke instance Anda.

Contoh ini didasarkan pada tumpukan Windows dengan satu lapisan khusus dan satu contoh. Ini menggunakan pencarian Chef dengan indeks `node` pencarian untuk mendapatkan alamat IP publik server dan menempatkan alamat dalam file di `C:\tmp` direktori. Berikut ini secara singkat merangkum cara membuat tumpukan untuk contoh ini. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan pilih Add Stack.
2. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan pilih Tambah Tumpukan.

   - Nama - NodeSearch
   - Wilayah - AS Barat (Oregon)

     Contoh ini akan bekerja di wilayah mana pun, tetapi kami sarankan menggunakan US West (Oregon) untuk tutorial.
   - Sistem operasi default - Microsoft Windows Server 2012 R2
3. Pilih Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

- Nama - IPTest
- Nama pendek — iptest

4. Tambahkan instance t2.micro 24/7 dengan pengaturan default ke IPTest layer dan mulai. Ini akan diberi nama iptest1.

   AWS OpsWorks Stacks secara otomatis menetapkan `AWS-OpsWorks-RDP-Server` ke instance ini, yang memungkinkan pengguna yang berwenang untuk masuk ke instance.

5. Pilih Izin dan kemudian Edit, dan pilih SSH/RDP dan sudo/admin. Pengguna reguler memerlukan otorisasi ini selain grup `AWS-OpsWorks-RDP-Server` keamanan untuk masuk ke instance.

> **Note**
>
> Anda juga dapat masuk sebagai Administrator, tetapi memerlukan prosedur yang berbeda. Untuk informasi selengkapnya, lihat Masuk dengan RDP.

Untuk mengatur buku masak

1. Buat direktori bernama `nodesearch` dan navigasikan ke sana.
2. Buat `metadata.rb` file dengan konten berikut dan simpan keopstest.

```
name "nodesearch"
version "0.1.0"
```

3. Buat `recipes` direktori di dalamnya`nodesearch`.
4. Buat `default.rb` file dengan resep berikut dan simpan ke `recipes` direktori.

```
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
```

Resepnya melakukan hal berikut:

1. Menggunakan sumber daya direktori untuk membuat `C:\tmp` direktori untuk file.

   Untuk informasi lebih lanjut tentang sumber daya ini, lihatContoh 3: Membuat Direktori.

2. Menggunakan pencarian Chef dengan indeks `node` pencarian untuk mendapatkan daftar node (instance) dengan nama host yang dimulai dengan. `iptest`

   Jika Anda menggunakan tema default, yang menciptakan nama host dengan menambahkan bilangan bulat ke nama pendek lapisan, kueri ini akan mengembalikan setiap instance di lapisan. IPTest Untuk contoh ini, lapisan diketahui hanya memiliki satu contoh, jadi resep hanya menetapkan yang pertama. `windowsserver` Untuk beberapa contoh, Anda bisa mendapatkan daftar lengkap dan kemudian menghitungnya.

3. Menambahkan pesan dengan alamat IP ke log Chef untuk proses ini.

   `windowsserver`Objek adalah tabel hash yang `ipaddress` atributnya diatur ke alamat IP publik instance, sehingga Anda dapat mewakili alamat tersebut dalam kode resep berikutnya sebagai`windowsserver[:ipaddress]`. Resep memasukkan string yang sesuai ke dalam pesan dan menambahkannya ke log Chef.

4. Menggunakan `file` sumber daya untuk membuat file dengan alamat IP bernama`C:\tmp\addresses.txt`.

   `content`Atribut sumber daya menentukan konten yang akan ditambahkan ke file, yang merupakan alamat IP publik dalam kasus ini.

5. Buat `.zip` arsip`nodesearch`, Unggah arsip ke bucket S3, buat arsip publik, dan rekam URL arsip.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya.

Untuk menginstal buku masak dan menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip S3
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

   Terima nilai default untuk pengaturan lain, dan pilih Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Perbarui Buku Masak Kustom, yang menginstal versi buku masak kustom Anda saat ini pada instance tumpukan, termasuk instance online. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.
3. Setelah Update Custom Cookbooks selesai, jalankan resep dengan menjalankan perintah stack Execute Recipes dengan Recipes untuk mengeksekusi set ke**`nodesearch::default`**. Perintah ini memulai menjalankan Chef, dengan daftar run yang terdiri dari resep Anda. Biarkan halaman execute_recipes terbuka.

Setelah resep berhasil berjalan, Anda dapat memverifikasinya.

Untuk memverifikasi nodesearch

1. Periksa log Chef untuk acara execute_recipes terbaru. Pada halaman Running command execute_recipes, pilih show di kolom Log instans iptest1 untuk menampilkan log. Gulir ke bawah untuk menemukan pesan log Anda di dekat bagian bawah, yang akan terlihat seperti berikut ini.

```
...
[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. Gunakan RDP untuk masuk ke instance dan memeriksa isi. `C:\tmp\addresses.txt`

Menggunakan Indeks Pencarian AWS OpsWorks Khusus Tumpukan di Tumpukan Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Contoh ini mengasumsikan bahwa Anda telah melakukan Menjalankan Resep pada Instance Windows contoh. Jika tidak, Anda harus melakukan contoh itu terlebih dahulu. Secara khusus, ini menjelaskan cara mengaktifkan akses RDP ke instance Anda.

AWS OpsWorks Stacks menyediakan indeks pencarian berikut selain: `node`

- `aws_opsworks_stack`— Konfigurasi tumpukan.
- `aws_opsworks_layer`— Konfigurasi lapisan tumpukan.
- `aws_opsworks_instance`— Konfigurasi instance tumpukan.
- `aws_opsworks_app`— Konfigurasi aplikasi tumpukan.
- `aws_opsworks_user`— Konfigurasi pengguna tumpukan.
- `aws_opsworks_rds_db_instance`— Informasi koneksi untuk instans RDS terdaftar.

Indeks ini mencakup beberapa atribut Chef standar, tetapi terutama ditujukan untuk mengambil atribut khusus AWS OpsWorks Stacks. Misalnya `aws_opsworks_instance` menyertakan `status` atribut yang menyediakan status instance, seperti`online`.

> **Note**
>
> Praktik yang disarankan adalah menggunakan node bila memungkinkan untuk menjaga resep Anda konsisten dengan penggunaan Chef standar. Sebagai contoh, lihat Menggunakan indeks pencarian node pada tumpukan Windows.

Contoh ini menunjukkan cara menggunakan indeks AWS OpsWorks Stacks untuk mengambil nilai atribut AWS OpsWorks khusus Stacks. Ini didasarkan pada tumpukan Windows sederhana dengan lapisan khusus yang memiliki satu contoh. Ini menggunakan pencarian Chef untuk mendapatkan ID AWS OpsWorks Stacks instance dan menempatkan hasilnya di log Chef.

Berikut ini secara singkat merangkum cara membuat tumpukan untuk contoh ini. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan pilih + Stack. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan pilih Tambah Tumpukan.

   - Nama - IDSearch
   - Wilayah - AS Barat (Oregon)

     Contoh ini akan bekerja di wilayah mana pun, tetapi kami sarankan menggunakan US West (Oregon) untuk tutorial.
   - Sistem operasi default - Microsoft Windows Server 2012 R2
2. Pilih Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama - IDCheck
   - Nama pendek - idcheck
3. Tambahkan instance t2.micro 24/7 dengan pengaturan default ke IDCheck layer dan mulai. Ini akan diberi nama iptest1.

   AWS OpsWorks Tumpukan secara otomatis menetapkan `AWS-OpsWorks-RDP-Server` untuk contoh ini. Mengaktifkan Akses RDPmenjelaskan cara menambahkan aturan masuk ke grup keamanan ini yang memungkinkan pengguna yang berwenang untuk masuk ke instance.

4. Pilih Izin dan kemudian Edit, dan pilih SSH/RDP dan sudo/admin. Pengguna reguler memerlukan otorisasi ini selain grup `AWS-OpsWorks-RDP-Server` keamanan untuk masuk ke instance.

> **Note**
>
> Anda juga dapat masuk sebagai Administrator, tetapi memerlukan prosedur yang berbeda. Untuk informasi selengkapnya, lihat Masuk dengan RDP.

Untuk mengatur buku masak

1. Buat direktori bernama `idcheck` dan navigasikan ke sana.
2. Buat `metadata.rb` file dengan konten berikut dan simpan keo`pstest`.

```
name "idcheck"
version "0.1.0"
```

3. Buat `recipes` direktori di dalam `idcheck` dan tambahkan `default.rb` file ke direktori yang berisi resep berikut.

```
windowsserver = search(:aws_opsworks_instance, "hostname:idcheck*").first
Chef::Log.info("**********The public IP address is:
 '#{windowsserver[:instance_id]}'**********")
```

Resepnya menggunakan pencarian Chef dengan indeks `aws_opsworks_instance` pencarian untuk mendapatkan atribut instance dari setiap instance di tumpukan dengan nama host yang dimulai dengan`idcheck`. Jika Anda menggunakan tema default, yang menciptakan nama host dengan menambahkan bilangan bulat ke nama pendek lapisan, kueri ini akan mengembalikan setiap instance di lapisan. IDCheck Untuk contoh ini, lapisan diketahui hanya memiliki satu contoh, jadi resep hanya menetapkan yang pertama. `windowsserver` Untuk beberapa contoh, Anda bisa mendapatkan daftar lengkap dan kemudian menghitungnya.

Resepnya memanfaatkan fakta bahwa hanya ada satu contoh di tumpukan dengan nama host ini, jadi hasil pertama adalah yang benar. Jika tumpukan Anda memiliki beberapa instance, pencarian pada atribut lain mungkin mengembalikan lebih dari satu hasil. Untuk daftar atribut instance, lihatTas Data Instance (aws_opsworks_instance).

Atribut instance pada dasarnya adalah tabel hash, dan ID AWS OpsWorks Stacks instance ditetapkan ke `instance_id` atribut, sehingga Anda dapat merujuk ke ID sebagai. `windowsserver[:instance_id]` Resep memasukkan string yang sesuai ke dalam pesan dan menambahkannya ke log Chef.

4. Buat `.zip` arsip `ipaddress` buku masak, Unggah arsip ke bucket Amazon S3, dan rekam URL arsip. Untuk informasi lebih lanjut tentang repositori buku masak, lihat. Repositori

   Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya.

Untuk menginstal buku masak dan menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip S3
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

   Terima nilai default untuk pengaturan lain, dan pilih Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Perbarui Buku Masak Kustom, yang menginstal versi buku masak kustom Anda saat ini pada instance tumpukan, termasuk instance online. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.
3. Setelah Update Custom Cookbooks selesai, jalankan resep dengan menjalankan perintah stack Execute Recipes dengan Recipes untuk mengeksekusi set ke**`idcheck::default`**. Perintah ini memulai menjalankan Chef, dengan daftar run yang terdiri dari resep Anda. Biarkan halaman execute_recipes terbuka.

Setelah resep berhasil berjalan, Anda dapat memverifikasinya dengan memeriksa log Chef untuk acara execute_recipes terbaru. Pada halaman Running command execute_recipes, pilih show di kolom Log instans iptest1 untuk menampilkan log. Gulir ke bawah untuk menemukan pesan log Anda di dekat bagian bawah, yang akan terlihat seperti berikut ini.

```
...
[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
...
```

Menggunakan Cookbook Eksternal pada Instance Linux: Berkshelf

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Berkshelf hanya tersedia untuk tumpukan Chef 11.10 Linux.

Sebelum Anda mulai menerapkan buku masak, lihat Buku Masak Komunitas Koki, yang berisi buku masak yang telah dibuat oleh anggota komunitas Chef untuk berbagai tujuan. Banyak dari buku masak ini dapat digunakan dengan AWS OpsWorks Stacks tanpa modifikasi, jadi Anda mungkin dapat memanfaatkannya untuk beberapa tugas Anda alih-alih menerapkan semua kode sendiri.

Untuk menggunakan buku masak eksternal pada sebuah instance, Anda memerlukan cara untuk menginstalnya dan mengelola dependensi apa pun. Pendekatan yang lebih disukai adalah menerapkan buku masak yang mendukung manajer ketergantungan bernama Berkshelf. Berkshelf berfungsi pada EC2 instans Amazon, termasuk instans AWS OpsWorks Stacks, tetapi juga dirancang untuk bekerja dengan Test Kitchen dan Vagrant. Namun, penggunaan pada Vagrant sedikit berbeda dengan AWS OpsWorks Stacks, jadi topik ini mencakup contoh untuk kedua platform. Untuk informasi selengkapnya tentang cara menggunakan Berkshelf, lihat Berkshelf.

Topik

- Menggunakan Berkshelf dengan Test Kitchen dan Vagrant
- Menggunakan Berkshelf dengan Stacks AWS OpsWorks

Menggunakan Berkshelf dengan Test Kitchen dan Vagrant

Contoh ini menunjukkan cara menggunakan Berkshelf untuk menginstal buku masak komunitas yang memulai dan menjalankan resepnya, yang menginstal file teks singkat di direktori home Anda pada instance.

Untuk menginstal Berkshelf dan menginisialisasi buku masak

1. Di workstation Anda, instal permata Berkshelf, sebagai berikut.

   ```
   gem install berkshelf
   ```

   Tergantung pada workstation Anda, perintah ini mungkin memerlukan`sudo`, atau Anda juga dapat menggunakan manajer lingkungan Ruby seperti RVM. Untuk memverifikasi bahwa Berkshelf berhasil diinstal, jalankan`berks --version`.
2. Buku masak untuk topik ini bernama external_cookbook. Anda dapat menggunakan Berkshelf untuk membuat buku masak yang diinisialisasi alih-alih pendekatan manual yang telah diambil topik sebelumnya. Untuk melakukannya, navigasikan ke `opsworks_cookbooks` direktori dan jalankan perintah berikut.

   ```
   berks cookbook external_cookbook
   ```

   Perintah membuat `external_cookbook` direktori dan beberapa subdirektori Chef dan Test Kitchen standar, termasuk `recipes` dan. `test` Perintah ini juga membuat versi default dari sejumlah file standar, termasuk yang berikut ini:

   - `metadata.rb`
   - File konfigurasi untuk Vagrant, Test Kitchen, dan Berkshelf
   - `default.rb`Resep kosong di `recipes` direktori

> **Note**
>
> Anda tidak perlu menjalankan`kitchen init`; `berks cookbook` perintah menangani tugas-tugas itu.

3. Jalankan `kitchen converge`. Buku masak yang baru dibuat tidak melakukan sesuatu yang menarik pada saat ini, tetapi buku itu menyatu.

> **Note**
>
> Anda juga dapat menggunakan `berks init` untuk menginisialisasi buku masak yang ada untuk menggunakan Berkshelf.

Untuk menggunakan Berkshelf untuk mengelola dependensi eksternal buku masak, direktori root buku masak harus berisi`Berksfile`, yang merupakan file konfigurasi yang menentukan bagaimana Berkshelf harus mengelola dependensi. Ketika Anda biasa `berks cookbook` membuat `external_cookbook` buku masak, itu dibuat `Berksfile` dengan konten berikut.

```
source "https://supermarket.chef.io"
metadata
```

File ini memiliki deklarasi berikut:

- `source`— URL sumber buku masak.

  Sebuah Berksfile dapat memiliki sejumlah `source` deklarasi, yang masing-masing menentukan sumber default untuk buku masak dependen. Jika Anda tidak secara eksplisit menentukan sumber buku masak, Berkshelf mencari di repositori default untuk buku masak dengan nama yang sama. Berksfile default mencakup `source` atribut tunggal yang menentukan repositori buku masak komunitas. Repositori itu berisi buku masak yang memulai, sehingga Anda dapat membiarkan baris tidak berubah.
- `metadata`— Mengarahkan Berkshelf untuk menyertakan dependensi buku masak yang dideklarasikan dalam file buku masak. `metadata.rb`

Anda juga dapat mendeklarasikan buku masak dependen di Berksfile dengan menyertakan `cookbook` atribut, seperti yang dibahas nanti.

Ada dua cara untuk mendeklarasikan ketergantungan buku masak:

- Dengan memasukkan `cookbook` deklarasi dalam Berksfile.

  Ini adalah pendekatan yang digunakan oleh AWS OpsWorks Stacks. Misalnya untuk menentukan buku masak memulai yang digunakan dalam contoh ini, sertakan `cookbook "getting-started"` dalam Berksfile. Berkshelf kemudian akan mencari di repositori default untuk buku masak dengan nama itu. Anda juga dapat menggunakan `cookbook` untuk secara eksplisit menentukan sumber buku masak, dan bahkan versi tertentu. Untuk informasi lebih lanjut, lihat Berkshelf.
- Dengan memasukkan `metadata` deklarasi di Berksfile dan mendeklarasikan ketergantungan di. `metadata.rb`

  Deklarasi ini mengarahkan Berkshelf untuk menyertakan dependensi buku masak yang dideklarasikan di. `metadata.rb` Misalnya, untuk mendeklarasikan dependensi memulai, tambahkan `depends 'getting-started'` deklarasi ke file buku masak. `metadata.rb`

Contoh ini menggunakan pendekatan pertama, untuk konsistensi dengan AWS OpsWorks Stacks.

Untuk menginstal buku masak yang memulai

1. Edit Berksfile default untuk mengganti `metadata` deklarasi dengan deklarasi untuk`cookbook`. `getting-started` Isinya akan terlihat seperti berikut.

   ```
   source "https://supermarket.chef.io"

   cookbook 'getting-started'
   ```

2. Jalankan`berks install`, yang mengunduh buku masak yang memulai dari repositori buku masak komunitas ke direktori Berkshelf workstation Anda, yang biasanya. `~/.berkshelf` Direktori ini sering disebut Berkshelf. Lihat di `cookbooks` direktori Berkshelf, dan Anda akan melihat direktori untuk buku masak yang memulai, yang akan diberi nama seperti. `getting-started-0.4.0`

3. Ganti `external_cookbook::default` dalam daftar `.kitchen.yml` run dengan`getting-started::default`. Contoh ini tidak menjalankan resep apa pun dari external_cookbook; ini pada dasarnya hanya cara untuk menggunakan buku masak yang memulai. `.kitchen.yml`File sekarang akan terlihat seperti berikut ini.

```
---
driver:
  name: vagrant

provisioner:
  name: chef_solo

platforms:
  - name: ubuntu-12.04

suites:
  - name: default
    run_list:
      - recipe[getting-started::default]
    attributes:
```

4. Jalankan `kitchen converge` dan kemudian gunakan `kitchen login` untuk masuk ke instance. Direktori login harus berisi file bernama `chef-getting-started.txt` dengan sesuatu seperti berikut:

```
Welcome to Chef!

This is Chef version 11.12.8.
Running on ubuntu.
Version 12.04.
```

   Test Kitchen menginstal buku masak di direktori instance. `/tmp/kitchen/cookbooks` Jika Anda mencantumkan isi direktori itu, Anda akan melihat dua buku masak: external_cookbook dan getting-started.

5. `kitchen destroy`Jalankan untuk mematikan instance. Contoh berikutnya menggunakan instance AWS OpsWorks Stacks.

Menggunakan Berkshelf dengan Stacks AWS OpsWorks

AWS OpsWorks Stacks secara opsional mendukung Berkshelf untuk tumpukan Chef 11.10. Untuk menggunakan Berkshelf dengan tumpukan Anda, Anda harus melakukan hal berikut.

- Aktifkan Berkshelf untuk tumpukan.

  AWS OpsWorks Stacks kemudian menangani rincian menginstal Berkshelf pada instance stack.
- Tambahkan Berksfile ke direktori root repositori buku masak Anda.

  Berksfile harus berisi `source` dan `cookbook` deklarasi untuk semua buku masak dependen.

Ketika AWS OpsWorks Stacks menginstal repositori buku masak kustom Anda pada sebuah instance, ia menggunakan Berkshelf untuk menginstal buku masak dependen yang dideklarasikan dalam Berksfile repositori. Untuk informasi selengkapnya, lihat Menggunakan Berkshelf.

Contoh ini menunjukkan cara menggunakan Berkshelf untuk menginstal buku masak komunitas yang memulai pada instance Stacks. AWS OpsWorks Ini juga menginstal versi createfile custom cookbook, yang membuat file dalam direktori tertentu. Untuk informasi selengkapnya tentang cara kerja createfile, lihat. Menginstal File dari Cookbook

> **Note**
>
> Jika ini adalah pertama kalinya Anda menginstal buku masak khusus pada tumpukan AWS OpsWorks Stacks, Anda harus terlebih dahulu melalui contoh. Menjalankan Resep pada Instance Linux

Mulailah dengan membuat tumpukan, seperti yang dirangkum dalam berikut ini. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan klik Add Stack.
2. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan klik Tambahkan Tumpukan.

   - Nama - BerksTest
   - Kunci SSH default - Sebuah EC2 key pair Amazon

Jika Anda perlu membuat EC2 key pair Amazon, lihat Amazon EC2 Key Pairs. Perhatikan bahwa key pair harus memiliki wilayah AWS yang sama dengan instans. Contoh menggunakan wilayah AS Barat (Oregon) default.

3. Klik Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama - BerksTest
   - Nama pendek — berkstest

   Anda benar-benar dapat menggunakan jenis lapisan apa pun untuk contoh ini. Namun, contoh tidak memerlukan paket apa pun yang diinstal oleh lapisan lain, jadi lapisan khusus adalah pendekatan yang paling sederhana.
4. Tambahkan instance 24/7 ke BerksTest layer dengan pengaturan default, tetapi jangan memulainya.

Dengan AWS OpsWorks Stacks, buku masak harus berada dalam repositori jarak jauh dengan struktur direktori standar. Anda kemudian memberikan informasi unduhan ke AWS OpsWorks Stacks, yang secara otomatis mengunduh repositori ke setiap instance tumpukan saat startup. Untuk mempermudah, repositori untuk contoh ini adalah arsip Amazon S3 publik, AWS OpsWorks tetapi Stacks juga mendukung arsip HTTP, repositori Git, dan repositori Subversion. Untuk informasi selengkapnya, lihat Repositori.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Untuk membuat repositori buku masak

1. Di `opsworks_cookbooks` direktori Anda, buat direktori bernama`berkstest_cookbooks`. Jika mau, Anda dapat membuat direktori ini di mana saja yang menurut Anda nyaman, karena Anda akan mengunggahnya ke repositori.
2. Tambahkan file bernama Berksfile `berkstest_cookbooks` dengan konten berikut.

```
source "https://supermarket.chef.io"
```

```
cookbook 'getting-started'
```

File ini mendeklarasikan ketergantungan buku masak yang memulai, dan mengarahkan Berkshelf untuk mengunduhnya dari situs buku masak komunitas.

3. Tambahkan `createfile` direktori `berkstest_cookbooks` yang berisi berikut ini.

   - `metadata.rb`File dengan konten berikut.

     ```
     name "createfile"
     version "0.1.0"
     ```

   - `files/default`Direktori yang berisi `example_data.json` file dengan konten berikut.

     ```
     {
       "my_name" : "myname",
       "your_name" : "yourname",
       "a_number" : 42,
       "a_boolean" : true
     }
     ```

     Nama dan konten file bersifat arbitrer. Resepnya cukup menyalin file ke lokasi yang ditentukan.

   - `recipes`Direktori yang berisi `default.rb` file dengan kode resep berikut.

     ```
     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
     ```

Resep ini membuat `/srv/www/shared` dan menyalin `example_data.json` ke direktori itu dari `files` direktori buku masak.

4. Buat `.zip` arsip`berkstest_cookbooks`, Unggah arsip ke bucket Amazon S3, buat arsip publik, dan rekam URL arsip.

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya.

Untuk menginstal buku masak dan menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus, dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip Http
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya
   - Kelola Berkshelf — Ya

   Dua pengaturan pertama memberi AWS OpsWorks Stacks informasi yang dibutuhkan untuk mengunduh repositori buku masak ke instans Anda. Pengaturan terakhir memungkinkan dukungan Berkshelf, yang mengunduh buku masak yang memulai ke instance. Terima nilai default untuk pengaturan lain dan klik Simpan untuk memperbarui konfigurasi tumpukan.
2. Edit BerksTest layer untuk menambahkan resep berikut ke acara siklus hidup Setup layer.

   - `getting-started::default`
   - `createfile::default`
3. Mulai instans. Peristiwa Setup terjadi setelah instance selesai booting. AWS OpsWorks Stacks kemudian menginstal repositori buku masak, menggunakan Berkshelf untuk mengunduh buku masak yang memulai, dan menjalankan pengaturan lapisan dan menyebarkan resep, termasuk dan. `getting-started::default` `createfile::default`
4. Setelah instance online, gunakan SSH untuk masuk. Anda harus melihat yang berikut

   - `/srv/www/shared`harus mengandung`example_data.json`.
   - `/root`harus mengandung`chef-getting-started.txt`.

   AWS OpsWorks Stacks menjalankan resep sebagai root, jadi memulai menginstal file di `/root` direktori daripada direktori home Anda.

Menggunakan SDK for Ruby: Mengunduh File dari Amazon S3

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Ada beberapa tugas, seperti berinteraksi dengan layanan AWS, yang tidak dapat ditangani dengan sumber daya Chef. Misalnya, terkadang lebih baik menyimpan file dari jarak jauh dan memiliki resep untuk mengunduhnya ke instance. Anda dapat menggunakan sumber daya remote_file untuk mengunduh file dari server jarak jauh. Namun, jika Anda ingin menyimpan file Anda di bucket Amazon S3, `remote_file` dapat mengunduh file-file itu hanya jika ACL memungkinkan operasi.

Resep dapat menggunakan AWS SDK untuk Rubyuntuk mengakses sebagian besar layanan AWS. Topik ini menunjukkan cara menggunakan SDK for Ruby untuk mengunduh file dari bucket S3.

> **Note**
>
> Untuk informasi selengkapnya tentang cara menggunakan enkripsi dan dekripsi AWS SDK untuk Rubyuntuk menangani, lihat AWS: :S3: :S3Object. Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Topik

- Menggunakan SDK for Ruby pada Instance Vagrant
- Menggunakan SDK for Ruby AWS OpsWorks pada Instans Stacks Linux
- Menggunakan SDK for Ruby AWS OpsWorks pada Instans Windows Stacks

Menggunakan SDK for Ruby pada Instance Vagrant

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini menjelaskan bagaimana resep yang berjalan pada instance Vagrant dapat menggunakan AWS SDK untuk Rubyuntuk mengunduh file dari Amazon S3. Sebelum memulai, Anda harus terlebih dahulu memiliki satu set AWS kredensial—kunci akses dan kunci akses rahasia—yang memungkinkan resep mengakses Amazon S3.

> **Important**
>
> Kami sangat menyarankan agar Anda tidak menggunakan kredensil akun root untuk tujuan ini. Sebagai gantinya, buat pengguna dengan kebijakan yang sesuai dan berikan kredensyal tersebut ke resep.
>
> Berhati-hatilah untuk tidak menempatkan kredensial—bahkan kredensil pengguna IAM—di lokasi yang dapat diakses publik, seperti dengan mengunggah file yang berisi kredensialnya ke repositori publik atau Bitbucket. GitHub Melakukan hal itu memperlihatkan kredensil Anda dan dapat membahayakan keamanan akun Anda.
>
> Resep yang berjalan pada EC2 instance EC2 Amazon dapat menggunakan pendekatan yang lebih baik, peran IAM, seperti yang dijelaskan dalamMenggunakan SDK for Ruby AWS OpsWorks pada Instans Stacks Linux.
>
> Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan.
>
> Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Jika Anda belum memiliki pengguna yang sesuai, Anda dapat membuatnya sebagai berikut. Untuk informasi lebih lanjut, lihat Apa itu IAM.

> **Warning**
>
> Pengguna IAM memiliki kredensyal jangka panjang, yang menghadirkan risiko keamanan. Untuk membantu mengurangi risiko ini, kami menyarankan agar Anda memberikan pengguna ini hanya izin yang mereka perlukan untuk melakukan tugas dan bahwa Anda menghapus pengguna ini ketika mereka tidak lagi diperlukan.

Untuk membuat pengguna IAM

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Pengguna dan, jika perlu, pilih Tambahkan pengguna untuk membuat pengguna administratif baru.
3. Pada halaman Setel izin, pilih Lampirkan kebijakan secara langsung.
4. Ketik **S3** kotak pencarian Kebijakan izin untuk menampilkan kebijakan Amazon S3.

   Pilih ReadOnlyAccessAmazonS3. Jika mau, Anda dapat menentukan kebijakan yang memberikan izin yang lebih luas, seperti AmazonS3 FullAccess, tetapi praktik standarnya adalah hanya memberikan izin yang diperlukan. Dalam hal ini, resepnya hanya akan mengunduh file, jadi akses hanya-baca sudah cukup.
5. Pilih Berikutnya.
6. Pilih Buat pengguna
7. Selanjutnya buat kunci akses untuk pengguna Anda. Untuk informasi selengkapnya tentang cara membuat access key, lihat Mengelola access key untuk pengguna IAM di Panduan Pengguna IAM.

Anda selanjutnya harus menyediakan file yang akan diunduh. Contoh ini mengasumsikan bahwa Anda akan meletakkan file bernama `myfile.txt` dalam bucket S3 yang baru dibuat bernama. `cookbook_bucket`

Untuk menyediakan file untuk diunduh

1. Buat file bernama `myfile.txt` dengan teks berikut dan simpan di lokasi yang nyaman di workstation Anda.

```
This is the file that you just downloaded from Amazon S3.
```

2. Di konsol Amazon S3, buat bucket bernama `cookbook_bucket` di wilayah Standar dan unggah `myfile.txt` ke bucket.

Atur buku masak sebagai berikut.

Untuk mengatur buku masak

1. Buat direktori dalam `opsworks_cookbooks` bernama `s3bucket` dan navigasikan ke sana.
2. Inisialisasi dan konfigurasikan Test Kitchen, seperti yang dijelaskan dalamContoh 1: Menginstal Paket.
3. Ganti teks `.kitchen.yml` dengan yang berikut ini.

```
---
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. Tambahkan dua direktori kes`3bucket`: `recipes` dan`environments`.
5. Buat file lingkungan bernama `test.json` dengan `default_attributes` bagian berikut, ganti `access_key` dan `secret_key` nilai dengan kunci yang sesuai untuk pengguna Anda. Simpan file ke `environments` folder buku masak.

```
{
  "default_attributes" : {
    "cookbooks_101" : {
      "access_key": "AKIAIOSFODNN7EXAMPLE",
      "secret_key" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
    }
  },
  "chef_type" : "environment",
  "json_class" : "Chef::Environment"
}
```

Anda memiliki berbagai cara untuk memberikan kredensyal ke resep yang berjalan pada sebuah instance. Pertimbangan utamanya adalah membatasi kemungkinan secara tidak sengaja mengekspos kunci dan membahayakan keamanan akun Anda. Untuk alasan itu, menggunakan nilai kunci eksplisit dalam kode Anda tidak disarankan. Contohnya malah menempatkan nilai-nilai kunci dalam objek node, yang memungkinkan resep untuk mereferensikannya dengan menggunakan sintaks node alih-alih mengekspos nilai literal. Anda harus memiliki hak akses root untuk mengakses objek node, yang membatasi kemungkinan bahwa kunci mungkin terbuka. Untuk informasi selengkapnya, lihat Praktik Terbaik untuk Mengelola Kunci Akses AWS.

> **Note**
>
> Perhatikan bahwa contoh menggunakan atribut bersarang, dengan `cookbooks_101` sebagai elemen pertama. Praktek ini membatasi kemungkinan tabrakan nama jika ada atribut lain `access_key` atau `secret_key` atribut dalam objek node.

Resep berikut diunduh `myfile.text` dari `cookbook_bucket` ember.

```
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
```

Bagian pertama dari resep menginstal SDK for Ruby, yang merupakan paket permata. Sumber daya gem_package menginstal permata yang akan digunakan oleh resep atau aplikasi lain.

> **Note**
>
> Instance Anda biasanya memiliki dua instance Ruby, yang biasanya merupakan versi yang berbeda. Salah satunya adalah instance khusus yang digunakan oleh klien Chef. Yang lain digunakan oleh aplikasi dan resep yang berjalan pada instance. Penting untuk memahami perbedaan ini saat menginstal paket permata, karena ada dua sumber daya untuk menginstal permata, gem_package dan chef_gem. Jika aplikasi atau resep menggunakan paket permata, instal dengan`gem_package`. `chef_gem`hanya untuk paket permata yang digunakan oleh klien Chef.

Sisa resep adalah sumber daya ruby_block, yang berisi kode Ruby yang mengunduh file. Anda mungkin berpikir bahwa karena resep adalah aplikasi Ruby, Anda dapat memasukkan kode ke dalam resep secara langsung. Namun, Chef run mengkompilasi semua kode itu sebelum mengeksekusi sumber daya apa pun. Jika Anda menempatkan kode contoh langsung dalam resep, Ruby akan mencoba untuk menyelesaikan `require 'aws-sdk'` pernyataan sebelum mengeksekusi sumber daya. `gem_package` Karena SDK for Ruby belum diinstal, kompilasi akan gagal.

Kode dalam `ruby_block` sumber daya tidak dikompilasi sampai sumber daya tersebut dijalankan. Dalam contoh ini, `ruby_block` sumber daya dieksekusi setelah `gem_package` sumber daya selesai menginstal SDK for Ruby, sehingga kode akan berjalan dengan sukses.

Kode dalam `ruby_block` karya sebagai berikut.

1. Membuat `Aws::S3`objek baru, yang menyediakan antarmuka layanan.

   Akses dan kunci rahasia ditentukan dengan mereferensikan nilai-nilai yang disimpan dalam objek node.

2. Memanggil `bucket.objects` asosiasi S3 objek, yang mengembalikan `Aws::S3::Object`objek bernama `myfile` yang mewakilimy`file.txt`.

3. Menggunakan `Dir.chdir` untuk mengatur direktori kerja ke`/tmp`.

4. Membuka file bernamamy`file.txt`, menulis isi `myfile` ke file, dan menutup file.

Untuk menjalankan resep

1. Buat file bernama `default.rb` dengan resep contoh dan simpan ke `recipes` direktori.
2. Jalankan `kitchen converge`.
3. Jalankan `kitchen login` untuk masuk ke instance, lalu jalankan`ls /tmp`. Anda akan melihatmy`file.txt`, bersama dengan beberapa file dan direktori Test Kitchen.

   ```
   vagrant@s3bucket-ubuntu-1204:~$ ls /tmp
   install.sh  kitchen  myfile.txt  stderr
   ```

   Anda juga dapat menjalankan `cat /tmp/myfile.txt` untuk memverifikasi bahwa konten file sudah benar.

Setelah selesai, jalankan `kitchen destroy` untuk menghentikan instance.

Menggunakan SDK for Ruby AWS OpsWorks pada Instans Stacks Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini menjelaskan cara menggunakan SDK for Ruby AWS OpsWorks pada instance Stacks Linux untuk mengunduh file dari bucket Amazon S3. AWS OpsWorks Stacks secara otomatis menginstal SDK for Ruby di setiap instance Linux. Namun, saat Anda membuat objek klien layanan, Anda harus menyediakan kumpulan kredensyal AWS yang sesuai `AWS::S3.new` atau yang setara untuk layanan lain.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Menggunakan SDK for Ruby pada Instance Vagrantmenunjukkan cara mengurangi risiko mengekspos kredensil Anda dengan menyimpan kredensil di objek node dan mereferensikan atribut dalam kode resep Anda. Saat Anda menjalankan resep di EC2 instans Amazon, Anda memiliki opsi yang lebih baik, peran IAM.

Peran IAM bekerja seperti pengguna IAM. Ini memiliki kebijakan terlampir yang memberikan izin untuk menggunakan berbagai layanan AWS. Namun, Anda menetapkan peran ke EC2 instans Amazon daripada individu. Aplikasi yang berjalan pada instance itu kemudian dapat memperoleh izin yang diberikan oleh kebijakan terlampir. Dengan peran, kredensyal tidak pernah muncul dalam kode Anda, bahkan secara tidak langsung. Topik ini menjelaskan bagaimana Anda dapat menggunakan peran IAM untuk menjalankan resep dari EC2 instans Menggunakan SDK for Ruby pada Instance Vagrant Amazon.

Anda dapat menjalankan resep ini dengan Test Kitchen menggunakan driver dapur-ec2, seperti yang dijelaskan dalam. Contoh 9: Menggunakan EC2 Instans Amazon Namun, menginstal SDK for Ruby EC2 di instans Amazon agak rumit dan bukan sesuatu yang perlu Anda perhatikan untuk Stacks. AWS OpsWorks Semua instance AWS OpsWorks Stacks Linux memiliki SDK for Ruby yang diinstal secara default. Untuk kesederhanaan, contoh karena itu menggunakan instance AWS OpsWorks Stacks.

Langkah pertama adalah mengatur peran IAM. Contoh ini mengambil pendekatan paling sederhana, yaitu menggunakan EC2 peran Amazon yang dibuat AWS OpsWorks Stacks saat Anda membuat tumpukan pertama Anda. Itu dinamai`aws-opsworks-ec2-role`. Namun, AWS OpsWorks Stacks tidak melampirkan kebijakan ke peran itu, jadi secara default tidak memberikan izin.

Anda harus melampirkan `AmazonS3ReadOnlyAccess` kebijakan ke `aws-opsworks-ec2-role` peran untuk memberikan izin yang sesuai. Untuk informasi selengkapnya tentang cara melampirkan kebijakan ke peran, lihat Menambahkan izin identitas (konsol) IAM di Panduan Pengguna IAM.

Anda menentukan peran saat membuat atau memperbarui tumpukan. Siapkan tumpukan dengan lapisan khusus, seperti yang dijelaskan dalamMenjalankan Resep pada Instance Linux, dengan satu tambahan. Pada halaman Add Stack, konfirmasikan bahwa profil instans IAM Default diatur ke aws-opsworks-ec2-role. AWS OpsWorks Tumpukan kemudian akan menetapkan peran itu ke semua instance tumpukan.

Prosedur untuk menyiapkan buku masak mirip dengan yang digunakan olehMenjalankan Resep pada Instance Linux. Berikut ini adalah ringkasan singkat; Anda harus merujuk pada contoh itu untuk detailnya.

Untuk mengatur buku masak

1. Buat direktori bernama `s3bucket_ops` dan navigasikan ke sana.
2. Buat `metadata.rb` file dengan konten berikut dan simpan kes`3bucket_ops`.

```
name "s3bucket_ops"
version "0.1.0"
```

3. Buat `recipes` direktori di dalamnyas`3bucket_ops`.
4. Buat `default.rb` file dengan resep berikut dan simpan ke `recipes` direktori.

```
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. Buat `.zip` arsip `s3bucket_ops` dan unggah arsip ke bucket Amazon S3. Untuk mempermudah, buat arsip publik, lalu rekam URL arsip untuk digunakan nanti. Anda juga dapat menyimpan buku masak Anda di arsip Amazon S3 pribadi, atau beberapa jenis repositori lainnya. Untuk informasi selengkapnya, lihat Repositori.

Resep ini mirip dengan yang digunakan oleh contoh sebelumnya, dengan pengecualian berikut.

- Karena AWS OpsWorks Stacks telah menginstal SDK for Ruby, sumber daya telah dihapus`chef_gem`.
- Resepnya tidak memberikan kredensi apa pun. `AWS::S3.new`

  Kredensil secara otomatis ditetapkan ke aplikasi berdasarkan peran instans.
- Resepnya digunakan `Chef::Log.info` untuk menambahkan pesan ke log Chef.

Buat tumpukan untuk contoh ini sebagai berikut. Anda juga dapat menggunakan tumpukan Windows yang ada. Cukup perbarui buku masak, seperti yang dijelaskan nanti.

Untuk membuat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan klik Add Stack.
2. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan klik Tambahkan Tumpukan.

   - Nama — RubySDK
   - Kunci SSH default - Sebuah EC2 key pair Amazon

   Jika Anda perlu membuat EC2 key pair Amazon, lihat Amazon EC2 Key Pairs. Perhatikan bahwa key pair harus memiliki wilayah AWS yang sama dengan instans. Contoh menggunakan wilayah AS Barat (Oregon) default.
3. Klik Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama — S3Download
   - Nama pendek — s3download

Tipe lapisan apa pun sebenarnya akan berfungsi untuk tumpukan Linux, tetapi contohnya tidak memerlukan paket apa pun yang diinstal oleh jenis lapisan lain, jadi lapisan khusus adalah pendekatan yang paling sederhana.

4. Tambahkan instance 24/7 dengan pengaturan default ke layer dan mulai.

Anda sekarang dapat menginstal dan menjalankan resep

Untuk menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus, dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip Http
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya.

   Gunakan nilai default untuk pengaturan lain dan klik Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Perbarui Buku Masak Kustom, yang menginstal versi buku masak kustom Anda saat ini pada instance tumpukan. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.
3. Jalankan resep dengan menjalankan perintah tumpukan Execute Recipes dengan Recipes untuk mengeksekusi set ke**`s3bucket_ops::default`**. Perintah ini memulai menjalankan Chef, dengan daftar run yang terdiri dari. `s3bucket_ops::default`

   > **Note**
   >
   > Anda biasanya memiliki AWS OpsWorks Stacks menjalankan resep Anda secara otomatis dengan menetapkannya ke acara siklus hidup yang sesuai. Anda dapat menjalankan resep tersebut dengan memicu acara secara manual. Anda dapat menggunakan perintah stack untuk memicu peristiwa Setup dan Configure, dan perintah deploy untuk memicu peristiwa Deploy dan Undeploy.

Setelah resep berjalan dengan sukses, Anda dapat memverifikasinya.

Untuk memverifikasi s3bucket_ops

1. Langkah pertama adalah memeriksa log Chef. Tumpukan Anda harus memiliki satu instance bernama opstest1. Pada halaman Instans, klik tampilkan di kolom Log instance untuk menampilkan log Chef. Gulir ke bawah dan untuk menemukan pesan log Anda di dekat bagian bawah.

```
...
[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. Gunakan SSH untuk masuk ke instance dan daftar isi. `/tmp`

Menggunakan SDK for Ruby AWS OpsWorks pada Instans Windows Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Contoh ini mengasumsikan bahwa Anda telah melakukan Menjalankan Resep pada Instance Windows contoh. Jika tidak, Anda harus melakukan contoh itu terlebih dahulu. Secara khusus, ini menjelaskan cara mengaktifkan akses RDP ke instance Anda.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Topik ini menjelaskan cara menggunakan instans Windows AWS OpsWorks Stacks untuk mengunduh file dari bucket S3. AWS SDK untuk Ruby

Jika aplikasi Ruby perlu mengakses sumber daya AWS, Anda harus menyediakannya dengan satu set kredensi AWS dengan izin yang sesuai. Untuk resep, opsi terbaik Anda untuk memberikan kredensil AWS adalah dengan menggunakan peran AWS Identity and Access Management (IAM). Peran IAM bekerja seperti pengguna IAM, ia memiliki kebijakan terlampir yang memberikan izin untuk menggunakan berbagai layanan. AWS Namun, Anda menetapkan peran ke instans Amazon Elastic Compute Cloud EC2 (Amazon), bukan ke individu. Aplikasi yang berjalan pada instance itu kemudian dapat memperoleh izin yang diberikan oleh kebijakan terlampir. Dengan peran, kredensyal tidak pernah muncul dalam kode Anda, bahkan secara tidak langsung.

Langkah pertama adalah mengatur peran IAM. Contoh ini mengambil pendekatan paling sederhana, yaitu menggunakan EC2 peran Amazon yang dibuat AWS OpsWorks Stacks saat Anda membuat tumpukan pertama Anda. Itu dinamai`aws-opsworks-ec2-role`. Namun, AWS OpsWorks Stacks tidak melampirkan kebijakan ke peran itu, jadi secara default tidak memberikan izin.

Anda harus melampirkan `AmazonS3ReadOnlyAccess` kebijakan ke `aws-opsworks-ec2-role` peran untuk memberikan izin yang sesuai. Untuk informasi selengkapnya tentang cara melampirkan kebijakan ke peran, lihat Menambahkan izin identitas (konsol) IAM di Panduan Pengguna IAM.

Anda menentukan peran saat membuat atau memperbarui tumpukan. Siapkan tumpukan dengan lapisan khusus, seperti yang dijelaskan dalamMenjalankan Resep pada Instance Windows, dengan satu tambahan. Pada halaman Add Stack, konfirmasikan bahwa profil instans IAM Default diatur ke aws-opsworks-ec2-role. AWS OpsWorks Tumpukan kemudian akan menetapkan peran itu ke semua instance tumpukan.

Prosedur untuk menyiapkan buku masak mirip dengan yang digunakan olehMenjalankan Resep pada Instance Linux. Berikut ini adalah ringkasan singkat; lihat contoh itu untuk detailnya.

Untuk mengatur buku masak

1. Buat direktori bernama `s3bucket_ops` dan navigasikan ke sana.
2. Buat `metadata.rb` file dengan konten berikut dan simpan kes`3bucket_ops`.

```
name "s3download"
version "0.1.0"
```

3. Buat `recipes` direktori di dalamnya`s3download`.
4. Buat `default.rb` file dengan resep berikut, dan simpan ke `recipes` direktori. Ganti *`windows-cookbooks`* dengan nama bucket S3 yang akan Anda gunakan untuk menyimpan file yang akan diunduh.

```
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. Buat `.zip` arsip `s3download` dan unggah file ke bucket S3. Jadikan file publik dan rekam URL untuk digunakan nanti.
6. Buat file teks bernama `myfile.txt` dan unggah ke bucket S3. Ini adalah file yang akan diunduh resep Anda, sehingga Anda dapat menggunakan ember yang nyaman.

Resepnya melakukan tugas-tugas berikut.

1: Instal SDK for Ruby v2.

Contoh menggunakan SDK for Ruby untuk men-download objek. Namun, AWS OpsWorks Stacks tidak menginstal SDK ini pada instance Windows, jadi bagian pertama resep menggunakan `chef_gem`sumber daya untuk menangani tugas itu. Anda menggunakan sumber daya ini untuk menginstal permata untuk digunakan oleh Chef, yang mencakup resep.

2: Unduh file.

Bagian ketiga dari resep menggunakan `ruby_block`sumber daya untuk menjalankan kode `myfile.txt` SDK for Ruby v2 untuk mengunduh dari *`windows-cookbooks`* bucket S3 yang diberi nama ke direktori instans. `/chef` *`windows-cookbooks`*Ubah nama bucket yang berisi`myfile.txt`.

> **Note**
>
> Resep adalah aplikasi Ruby, jadi Anda dapat memasukkan kode Ruby ke dalam badan resep; itu tidak harus ada di sumber daya. `ruby_block` Namun, Chef mengeksekusi kode Ruby di tubuh resep terlebih dahulu, diikuti oleh setiap sumber daya, secara berurutan. Untuk contoh ini, jika Anda meletakkan kode unduhan di badan resep, itu akan gagal karena tergantung pada SDK for Ruby, `chef_gem` dan sumber daya yang menginstal SDK belum dieksekusi. Kode dalam `ruby_block` sumber daya dijalankan ketika sumber daya dijalankan, dan itu terjadi setelah `chef_gem` sumber daya menginstal SDK for Ruby.

Buat tumpukan untuk contoh ini sebagai berikut. Anda juga dapat menggunakan tumpukan Windows yang ada. Cukup perbarui buku masak, seperti yang dijelaskan nanti.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan pilih Add Stack. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan pilih Tambah Tumpukan.

   - Nama — S3Download
   - Wilayah - AS Barat (Oregon)

     Contoh ini akan bekerja di wilayah mana pun, tetapi kami sarankan menggunakan US West (Oregon) untuk tutorial.
   - Sistem operasi default - Microsoft Windows Server 2012 R2

2. Pilih Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama — S3Download
   - Nama pendek — s3download

3. Tambahkan instance 24/7 dengan pengaturan default ke layer S3Download dan mulai.

Anda sekarang dapat menginstal dan menjalankan resep

Untuk menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus, dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip S3.
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya.

   Terima nilai default untuk pengaturan lain dan pilih Simpan untuk memperbarui konfigurasi tumpukan.

2. Jalankan perintah tumpukan Update Custom Cookbooks, yang menginstal versi terbaru dari buku masak kustom Anda pada instance online stack. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.
3. Jalankan resep dengan menjalankan perintah tumpukan Execute Recipes dengan Recipes untuk mengeksekusi set ke**`s3download::default`**. Perintah ini memulai menjalankan Chef, dengan daftar run yang terdiri dari. `s3download::default`

   > **Note**
   >
   > Anda biasanya memiliki AWS OpsWorks Stacks menjalankan resep Anda secara otomatis dengan menetapkannya ke acara siklus hidup yang sesuai. Anda juga dapat menjalankan resep tersebut dengan memicu acara secara manual. Anda dapat menggunakan perintah stack untuk memicu peristiwa Setup dan Configure, dan perintah deploy untuk memicu peristiwa Deploy dan Undeploy.

Setelah resep berjalan dengan sukses, Anda dapat memverifikasinya.

Untuk memverifikasi s3download

1. Langkah pertama adalah memeriksa log Chef. Tumpukan Anda harus memiliki satu instance bernama s3download1. Pada halaman Instans, pilih tampilkan di kolom Log instance untuk menampilkan log Chef. Gulir ke bawah untuk menemukan pesan log Anda di dekat bagian bawah.

```
...
[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. Gunakan RDP untuk masuk ke instance dan memeriksa isi. `c:\chef`

Menginstal Perangkat Lunak Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Contoh-contoh ini mengasumsikan bahwa Anda telah melakukan Menjalankan Resep pada Instance Windows contoh. Jika tidak, Anda harus melakukan contoh itu terlebih dahulu. Secara khusus, ini menjelaskan cara mengaktifkan akses RDP ke instance Anda.

Instans Windows dimulai dengan Windows Server 2012 R2 Standard, jadi Anda biasanya perlu menginstal beberapa perangkat lunak. Detailnya tergantung pada jenis perangkat lunak.

- Fitur Windows adalah komponen sistem opsional, termasuk kerangka kerja .NET dan Layanan Informasi Internet (IIS), yang dapat Anda unduh ke instans Anda.
- Perangkat lunak pihak ketiga biasanya datang dalam paket installer, seperti file MSI, yang harus Anda unduh ke instance dan kemudian jalankan.

  Beberapa perangkat lunak Microsoft juga hadir dalam paket installer.

Bagian ini menjelaskan cara menerapkan buku masak untuk menginstal fitur dan paket Windows. Ini juga memperkenalkan Chef windows cookbook, yang berisi sumber daya dan fungsi pembantu yang menyederhanakan penerapan resep untuk instance Windows.

Topik

- Menginstal Fitur Windows: IIS
- Menginstal Package pada Instance Windows

Menginstal Fitur Windows: IIS

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Fitur Windows adalah seperangkat komponen sistem opsional, termasuk kerangka kerja .NET dan Layanan Informasi Internet (IIS). Topik ini menjelaskan bagaimana menerapkan buku masak untuk menginstal fitur yang umum digunakan, Internet Information Services (IIS).

> **Note**
>
> Menginstal Packagemenunjukkan cara menginstal perangkat lunak yang datang dalam paket installer, seperti file MSI, yang harus Anda unduh ke instance dan jalankan. Buku masak IIS

Menjalankan Resep pada Instance Windowsmenunjukkan cara menggunakan sumber `powershell_script` daya untuk menginstal fitur Windows. Contoh ini menunjukkan pendekatan alternatif: gunakan `windows_feature` sumber daya buku masak Chef Windows. Buku masak ini berisi sekumpulan sumber daya yang menggunakan Deployment Image Servicing and Management untuk melakukan berbagai tugas di Windows, termasuk instalasi fitur.

> **Note**
>
> Chef juga memiliki buku masak IIS, yang dapat Anda gunakan untuk mengelola IIS. Untuk informasi lebih lanjut, lihat buku masak IIS.

Untuk mengatur buku masak

1. Buka GitHub repositori buku masak windows dan unduh buku masak. `windows`

   Contoh ini mengasumsikan bahwa Anda akan mengunduh `windows` repositori sebagai file.zip, tetapi Anda juga dapat mengkloning repositori jika Anda mau.
2. Buka GitHub repositori buku masak chef_handler dan unduh buku masak. `chef-handler`

   `windows`Buku masak tergantung pada`chef_handler`; Anda tidak akan menggunakannya secara langsung. Contoh ini mengasumsikan bahwa Anda akan mengunduh `chef_handler` repositori sebagai file.zip, tetapi Anda juga dapat mengkloning repositori jika Anda mau.
3. Ekstrak `chef_handler` buku masak `windows` dan ke direktori di direktori buku masak Anda bernama `windows` dan`chef_handler`, masing-masing.
4. Buat direktori di direktori buku masak Anda bernama `install-iis` dan arahkan ke sana.
5. Tambahkan `metadata.rb` file `install-iis` dengan konten berikut.

   ```
   name "install-iis"
   version "0.1.0"

   depends "windows"
   ```

   `depends`Arahan ini memungkinkan Anda untuk menggunakan sumber daya `windows` buku masak dalam resep Anda.
6. Tambahkan `recipes` direktori ke `install-iis` dan tambahkan file bernama `default.rb` ke direktori yang berisi kode resep berikut.

```
%w{ IIS-WebServerRole IIS-WebServer }.each do |feature|
  windows_feature feature do
    action :install
  end
end

service 'w3svc' do
  action [:start, :enable]
end
```

Resepnya menggunakan `windows_feature` sumber daya `windows` buku masak untuk menginstal yang berikut ini:

1. Peran Server Web IIS.
2. Server Web IIS.

Resep kemudian menggunakan `service`sumber daya untuk memulai dan mengaktifkan layanan IIS (W3SVC).

> **Note**
>
> Untuk daftar lengkap fitur Windows yang tersedia, gunakan RDP untuk masuk ke instance, buka jendela prompt perintah, dan jalankan perintah berikut. Perhatikan bahwa daftarnya cukup panjang.
>
> ```
> dism /online /Get-Features
> ```

7. Buat `.zip` arsip yang berisi`install-iis`,`chef_handler`, dan `windows` buku masak dan unggah arsip ke ember S3. Jadikan arsip publik dan rekam URL untuk digunakan nanti. Contoh ini mengasumsikan bahwa arsip diberi nama`install-iis.zip`. Untuk informasi selengkapnya, lihat Repositori.

   Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Buat tumpukan untuk contoh ini sebagai berikut. Anda juga dapat menggunakan tumpukan Windows yang ada. Cukup perbarui buku masak, seperti yang dijelaskan nanti.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan pilih Add Stack. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan pilih Tambah Tumpukan.

   - Nama — Installiis
   - Wilayah - AS Barat (Oregon)

     Contoh ini akan bekerja di wilayah mana pun, tetapi kami sarankan menggunakan US West (Oregon) untuk tutorial.
   - Sistem operasi default - Microsoft Windows Server 2012 R2
2. Pilih Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama — IIS
   - Nama pendek — iis
3. Tambahkan instance 24/7 dengan pengaturan default ke lapisan IIS dan mulailah.

Anda sekarang dapat menginstal buku masak dan menjalankan resepnya

Untuk menginstal buku masak dan menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus, dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip S3
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya.

   Terima nilai default untuk pengaturan lain dan pilih Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Update Custom Cookbooks, yang menginstal versi terbaru buku masak kustom Anda pada instance online stack. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.

3. Jalankan resep dengan menjalankan perintah tumpukan Execute Recipes dengan Recipes untuk mengeksekusi set ke**`install-iis::default`**. Perintah ini memulai menjalankan Chef, yang menjalankan resep yang ditentukan.

   > **Note**
   >
   > Contoh ini menggunakan Execute Recipes untuk kenyamanan, tetapi Anda biasanya memiliki AWS OpsWorks Stacks menjalankan resep Anda secara otomatis dengan menetapkannya ke acara siklus hidup yang sesuai. Anda dapat menjalankan resep tersebut dengan memicu acara secara manual. Anda dapat menggunakan perintah stack untuk memicu peristiwa Setup dan Configure, dan perintah deploy untuk memicu peristiwa Deploy dan Undeploy.

4. Untuk memverifikasi instalasi, gunakan RDP untuk terhubung ke instance dan buka Windows Explorer. Sistem file sekarang harus memiliki `C:\inetpub` direktori. Jika Anda memeriksa daftar layanan di aplikasi Panel Kontrol Alat Administratif, IIS harus berada di dekat bagian bawah. Namun, itu akan dinamai World Wide Web Publishing Service, bukan IIS.

Menginstal Package pada Instance Windows

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Contoh ini mengasumsikan bahwa Anda telah melakukan Menjalankan Resep pada Instance Windows contoh. Jika tidak, Anda harus melakukan contoh itu terlebih dahulu. Secara khusus, ini menjelaskan cara mengaktifkan akses RDP ke instance Anda.

Jika perangkat lunak Anda datang dalam paket installer, seperti MSI, Anda harus mengunduh file ke instance dan kemudian menjalankannya. Contoh ini menunjukkan bagaimana menerapkan buku masak untuk menginstal paket MSI, runtime Python, termasuk cara mendefinisikan variabel lingkungan terkait. Untuk informasi selengkapnya tentang cara menginstal fitur Windows seperti IIS, lihatMenginstal Fitur Windows: IIS.

Untuk mengatur buku masak

1. Buat direktori bernama `installpython` dan navigasikan ke sana.
2. Tambahkan `metadata.rb` file `installpython` dengan konten berikut.

```
name "installpython"
version "0.1.0"
```

3. Tambahkan `recipes` dan `files` direktori ke `installpython` dan tambahkan `default` direktori ke file.
4. Unduh paket Python dari Rilis Python untuk Windows ke direktori buku masak. `files\default` Contoh ini menginstal versi Windows x86-64 dari Python 3.5.0a3, yang menggunakan installer MSI bernama. `python-3.4.3.amd64.msi`
5. Tambahkan file bernama `default.rb` ke `recipes` direktori dengan kode resep berikut.

```
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
```

Resepnya melakukan hal berikut:

1. Menggunakan sumber daya direktori untuk membuat `C:\tmp` direktori.

   Untuk informasi lebih lanjut tentang sumber daya ini, lihatContoh 3: Membuat Direktori.

2. Menggunakan sumber daya cookbook_file untuk menyalin penginstal dari direktori buku masak ke. `files\default` `C:\tmp`

   Untuk informasi lebih lanjut tentang sumber daya ini, lihatMenginstal File dari Cookbook.

3. Menggunakan sumber daya windows_package untuk menjalankan installer MSI, yang menginstal Python. `c:\python34`

   Installer membuat direktori yang diperlukan dan menginstal file, tetapi tidak memodifikasi variabel lingkungan sistem. `PATH`

4. Menggunakan sumber daya env untuk ditambahkan `c:\python34` ke jalur sistem.

   Anda menggunakan sumber daya env untuk menentukan variabel lingkungan. Dalam hal ini, resep memungkinkan Anda untuk dengan mudah menjalankan skrip Python dari baris perintah dengan menambahkan `c:\python34` ke jalur.

   - Nama sumber daya menentukan nama variabel lingkungan, `PATH` untuk contoh ini.
   - `value`Atribut menentukan nilai variabel, `c:\\python34` untuk contoh ini (Anda perlu untuk melarikan diri dari `\` karakter).
   - `:modify`Tindakan menambahkan nilai yang ditentukan ke nilai variabel saat ini.
   - `delim`Atribut menentukan pembatas yang memisahkan nilai baru dari nilai yang ada, yang untuk contoh ini`;`.

6. Buat `.zip` arsip`installpython`, unggah arsip ke bucket S3, dan buat publik. Rekam URL arsip untuk digunakan nanti. Untuk informasi selengkapnya, lihat Repositori.

   Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Buat tumpukan untuk contoh ini sebagai berikut. Anda juga dapat menggunakan tumpukan Windows yang ada. Cukup perbarui buku masak, seperti yang dijelaskan nanti.

Buat tumpukan

1. Buka konsol AWS OpsWorks Stacks dan pilih Add Stack. Tentukan pengaturan berikut, terima default untuk pengaturan lainnya, dan pilih Tambah Tumpukan.

   - Nama - InstallPython
   - Wilayah - AS Barat (Oregon)

     Contoh ini akan bekerja di wilayah mana pun, tetapi kami sarankan menggunakan US West (Oregon) untuk tutorial.
   - Sistem operasi default - Microsoft Windows Server 2012 R2
2. Pilih Tambahkan lapisan dan tambahkan lapisan khusus ke tumpukan dengan pengaturan berikut.

   - Nama - Python
   - Nama pendek - python
3. Tambahkan instance 24/7 dengan pengaturan default ke lapisan Python dan mulai.

Setelah instance online, Anda dapat menginstal buku masak dan menjalankan resepnya

Untuk menginstal buku masak dan menjalankan resep

1. Edit tumpukan untuk mengaktifkan buku masak khusus, dan tentukan pengaturan berikut.

   - Jenis repositori - Arsip S3.
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya.

   Terima nilai default untuk pengaturan lain dan pilih Simpan untuk memperbarui konfigurasi tumpukan.
2. Jalankan perintah tumpukan Update Custom Cookbooks, yang menginstal versi terbaru buku masak kustom Anda pada instance online stack. Jika versi sebelumnya dari buku masak Anda ada, perintah ini menimpanya.

3. Jalankan resep dengan menjalankan perintah tumpukan Execute Recipes dengan Recipes untuk mengeksekusi set ke**`installpython::default`**. Perintah ini memulai menjalankan Chef, dengan daftar run yang terdiri dari. `installpython::default`

   > **Note**
   >
   > Contoh ini menggunakan Execute Recipes untuk kenyamanan, tetapi Anda biasanya memiliki AWS OpsWorks Stacks menjalankan resep Anda secara otomatis dengan menetapkannya ke acara siklus hidup yang sesuai. Anda dapat menjalankan resep tersebut dengan memicu acara secara manual. Anda dapat menggunakan perintah stack untuk memicu peristiwa Setup dan Configure, dan perintah deploy untuk memicu peristiwa Deploy dan Undeploy.

4. Untuk memverifikasi instalasi, gunakan RDP untuk terhubung ke instance dan buka Windows Explorer.

   - Sistem file sekarang harus memiliki `C:\Python34` direktori.
   - Jika Anda menjalankan `path` dari baris perintah, itu akan terlihat seperti: `PATH=c:\python34;C:\Windows\system32;...`
   - Jika Anda menjalankan `python --version` dari baris perintah, itu akan kembali`Python 3.4.3`.

Mengesampingkan Atribut Bawaan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Topik ini hanya berlaku untuk tumpukan Linux. Anda tidak dapat mengganti atribut bawaan pada tumpukan Windows.

AWS OpsWorks Stacks menginstal satu set buku masak bawaan pada setiap instance. Banyak buku masak bawaan mendukung lapisan bawaan, dan file atributnya menentukan berbagai pengaturan sistem dan aplikasi default, seperti pengaturan konfigurasi server Apache. Dengan menempatkan pengaturan ini dalam file atribut, Anda dapat menyesuaikan banyak pengaturan konfigurasi dengan mengganti atribut bawaan yang sesuai dengan salah satu cara berikut:

- Tentukan atribut di JSON kustom.

  Pendekatan ini memiliki keuntungan menjadi sederhana dan fleksibel. Namun, Anda harus memasukkan JSON khusus secara manual, sehingga tidak ada cara yang kuat untuk mengelola definisi atribut.
- Menerapkan buku masak kustom dan menentukan atribut dalam file `customize.rb` atribut.

  Pendekatan ini kurang fleksibel daripada menggunakan JSON khusus, tetapi lebih kuat karena Anda dapat menempatkan buku masak khusus di bawah kendali sumber.

Topik ini menjelaskan cara menggunakan file atribut buku masak kustom untuk mengganti atribut bawaan, menggunakan server Apache sebagai contoh. Untuk informasi selengkapnya tentang cara mengganti atribut dengan JSON kustom, lihat. Menggunakan JSON Kustom Untuk diskusi umum tentang cara mengganti atribut, lihatMengesampingkan Atribut.

> **Note**
>
> Mengganti atribut adalah cara yang lebih disukai untuk menyesuaikan pengaturan konfigurasi, tetapi pengaturan tidak selalu diwakili oleh atribut. Dalam hal ini, Anda sering dapat menyesuaikan file konfigurasi dengan mengganti template yang digunakan resep bawaan untuk membuat file konfigurasi. Sebagai contoh, lihat Mengesampingkan Template Bawaan.

Atribut bawaan biasanya mewakili nilai dalam file template yang digunakan resep Setup untuk membuat file konfigurasi. Misalnya, salah satu resep `apache2` Setup, `default.rb`, menggunakan

apache2.conf.erbtemplate untuk membuat file konfigurasi utama server Apache, `httpd.conf` (Amazon Linux) atau `apache2.conf` (Ubuntu). Berikut ini adalah kutipan dari file template:

```
...
#
# 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)
##

...
```

`KeepAliveTimeout`Pengaturan dalam contoh ini adalah nilai `[:apache][:keepalivetimeout]` atribut. Nilai default atribut ini didefinisikan dalam file apache.rbatribut `apache2` buku masak, seperti yang ditunjukkan dalam kutipan berikut:

```
...
# 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**
>
> Untuk informasi selengkapnya tentang atribut bawaan yang umum digunakan, lihatAtribut Buku Masak Bawaan.

Untuk mendukung penggantian atribut bawaan, semua buku masak bawaan berisi file `customize.rb` atribut, yang dimasukkan ke dalam semua modul melalui arahan. `include_attribute` `customize.rb`File buku masak bawaan tidak mengandung definisi atribut dan tidak berpengaruh pada atribut bawaan. Untuk mengganti atribut bawaan, Anda membuat buku masak kustom dengan nama yang sama dengan buku masak bawaan dan menempatkan definisi atribut kustom Anda dalam file atribut yang juga diberi nama. `customize.rb` File itu lebih diutamakan daripada versi bawaan, dan disertakan dalam modul terkait apa pun. Jika Anda menentukan atribut bawaan apa pun di Anda`customize.rb`, atribut tersebut akan mengganti atribut bawaan yang sesuai.

Contoh ini menunjukkan cara mengganti `[:apache][:keepalivetimeout]` atribut bawaan untuk menetapkan nilainya menjadi 5, bukan 3. Anda dapat menggunakan pendekatan serupa untuk atribut bawaan apa pun. Namun, berhati-hatilah atribut mana yang Anda timpa. Misalnya, mengganti atribut di `opsworks` namespace dapat menyebabkan masalah untuk beberapa resep bawaan.

> **Important**
>
> Jangan mengganti atribut bawaan dengan memodifikasi salinan file atribut bawaan itu sendiri. Misalnya, Anda dapat meletakkan salinan `apache.rb` di `apache2/attributes` folder buku masak khusus Anda dan memodifikasi beberapa pengaturannya. Namun, file ini lebih diutamakan daripada versi bawaan, dan resep bawaan sekarang akan menggunakan versi Anda. `apache.rb` Jika AWS OpsWorks Stacks kemudian memodifikasi `apache.rb` file bawaan, resep tidak akan mendapatkan nilai baru kecuali Anda memperbarui versi Anda secara manual. Dengan menggunakan`customize.rb`, Anda hanya mengganti atribut yang ditentukan; resep bawaan terus secara otomatis mendapatkan up-to-date nilai untuk setiap atribut yang belum Anda ganti.

Untuk memulai, buat buku masak khusus.

Untuk membuat buku masak

1. Di dalam `opsworks_cookbooks` direktori Anda, buat direktori buku masak bernama `apache2` dan arahkan ke sana.

   Untuk mengganti atribut bawaan, buku masak khusus harus memiliki nama yang sama dengan buku masak bawaan, `apache2` untuk contoh ini.
2. Di `apache2` direktori, buat `attributes` direktori.
3. Tambahkan file bernama `customize.rb` ke `attributes` direktori dan gunakan untuk menentukan atribut buku masak bawaan yang ingin Anda timpa. Untuk contoh ini, file harus berisi yang berikut:

   ```
   normal[:apache][:keepalivetimeout] = 5
   ```

   > **Important**
   >
   > Untuk mengganti atribut bawaan, atribut kustom harus berupa `normal` tipe atau lebih tinggi dan memiliki nama node yang sama persis dengan atribut bawaan yang sesuai. `normal`Tipe memastikan bahwa atribut kustom lebih diutamakan daripada atribut bawaan, yang semuanya `default` tipe. Untuk informasi selengkapnya, lihat Atribut Prioritas.
4. Buat `.zip` arsip `opsworks_cookbooks` bernama `opsworks_cookbooks.zip` dan unggah arsip ke bucket Amazon Simple Storage Service (Amazon S3). Untuk kesederhanaan, buat file publik. Rekam URL untuk digunakan nanti. Anda juga dapat menyimpan buku masak Anda di arsip Amazon S3 pribadi atau dalam jenis repositori lainnya. Untuk informasi selengkapnya, lihat Repositori.

   Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

Untuk menggunakan atribut kustom, buat tumpukan dan instal buku masak.

Untuk menggunakan atribut kustom

1. Buka konsol AWS OpsWorks Stacks, lalu pilih Add Stack.

2. Tentukan pengaturan standar berikut.

   - Nama - ApacheConfig
   - Wilayah - AS Barat (Oregon)

     Anda dapat menempatkan tumpukan Anda di wilayah mana pun, tetapi kami merekomendasikan US West (Oregon) untuk tutorial.
   - Kunci SSH default - Sebuah EC2 key pair

     Jika Anda perlu membuat EC2 key pair, lihat Amazon EC2 Key Pairs. Perhatikan bahwa key pair harus memiliki wilayah AWS yang sama dengan stack.

   Pilih Advanced>>, atur Gunakan buku masak Chef kustom ke Ya, lalu tentukan pengaturan berikut.

   - Jenis repositori - Arsip Http
   - URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

   Terima default untuk pengaturan lain, lalu pilih Tambah Tumpukan untuk membuat tumpukan.

   > **Note**
   >
   > Contoh ini menggunakan sistem operasi default, Amazon Linux. Anda dapat menggunakan Ubuntu, jika Anda mau. Satu-satunya perbedaan adalah bahwa pada sistem Ubuntu, resep Pengaturan bawaan menghasilkan file konfigurasi dengan pengaturan yang sama bernama `apache2.conf` dan meletakkannya di `/etc/apache2` direktori.

3. Pilih Tambahkan lapisan, lalu tambahkan lapisan Java App Server dengan pengaturan default ke tumpukan.
4. Tambahkan instance 24/7 dengan pengaturan default ke layer, lalu mulai instance.

   Sebuah instance t2.micro sudah cukup untuk contoh ini.
5. Setelah instance online, hubungkan dengan SSH. `httpd.conf`File ada di `/etc/httpd/conf` direktori. Jika Anda memeriksa file, Anda akan melihat `KeepAliveTimeout` pengaturan kustom Anda. Sisa pengaturan akan memiliki nilai default dari `apache.rb` file bawaan. Bagian yang relevan `httpd.conf` harus terlihat mirip dengan yang berikut:

```
...
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
...
```

Mengesampingkan Template Bawaan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Topik ini hanya berlaku untuk tumpukan Linux. Anda tidak dapat mengganti templat bawaan di tumpukan Windows.

Resep bawaan AWS OpsWorks Stacks menggunakan templat untuk membuat file pada instance, terutama file konfigurasi untuk server, seperti Apache. Misalnya, `apache2` resep menggunakan `apache2.conf.erb`template untuk membuat file konfigurasi utama server Apache, `httpd.conf` (Amazon Linux) atau `apache2.conf` (Ubuntu).

Sebagian besar pengaturan konfigurasi dalam template ini diwakili oleh atribut, jadi cara yang lebih disukai untuk menyesuaikan file konfigurasi adalah dengan mengganti atribut bawaan yang sesuai. Sebagai contoh, lihat Mengesampingkan Atribut Bawaan. Namun, jika pengaturan yang ingin Anda sesuaikan tidak diwakili oleh atribut bawaan, atau tidak ada dalam template sama sekali, Anda harus mengganti template itu sendiri. Topik ini menjelaskan cara mengganti template bawaan untuk menentukan pengaturan konfigurasi Apache kustom.

Anda dapat memberikan respons kesalahan kustom ke Apache dengan menambahkan `ErrorDocument` pengaturan ke `httpd.conf` file. `apache2.conf.erb`hanya berisi beberapa contoh yang dikomentari, seperti yang ditunjukkan dalam berikut ini:

```
...
#
# 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
...
```

Karena pengaturan ini adalah komentar hardcode, Anda tidak dapat menentukan nilai kustom dengan mengganti atribut; Anda harus mengganti template itu sendiri. Namun, tidak seperti atribut, tidak ada cara untuk mengganti bagian tertentu dari file template. Anda harus membuat buku masak khusus dengan nama yang sama dengan versi bawaan, menyalin file template ke subdirektori yang sama, dan memodifikasi file sesuai kebutuhan. Topik ini menunjukkan cara mengganti `apache2.conf.erb` untuk memberikan respons khusus terhadap kesalahan 500. Untuk diskusi umum tentang template utama, lihatMenggunakan Template Kustom.

> **Important**
>
> Saat Anda mengganti template Buiit-in, resep bawaan menggunakan versi template yang disesuaikan alih-alih versi bawaan. Jika AWS OpsWorks Stacks memperbarui template bawaan, template kustom menjadi tidak sinkron dan mungkin tidak berfungsi dengan benar. AWS OpsWorks Tumpukan tidak sering membuat perubahan seperti itu, dan ketika template berubah, AWS OpsWorks Stacks mencantumkan perubahan dan memberi Anda opsi untuk meningkatkan ke versi baru. Kami menyarankan Anda memantau repositori AWS OpsWorks Stacks untuk perubahan, dan memperbarui template kustom Anda secara manual sesuai kebutuhan. Perhatikan bahwa repositori memiliki cabang terpisah untuk setiap versi Chef yang didukung, jadi pastikan Anda berada di cabang yang benar.

Untuk memulai, buat buku masak khusus.

Untuk membuat buku masak

1. Di `opsworks_cookbooks` direktori, buat direktori buku masak bernama`apache2`, lalu arahkan ke sana. Untuk mengganti templat bawaan, buku masak khusus harus memiliki nama yang sama dengan buku masak bawaan, `apache2` untuk contoh ini.

   > **Note**
   >
   > Jika Anda telah menyelesaikan Mengesampingkan Atribut Bawaan penelusuran, Anda dapat menggunakan `apache2` buku masak yang sama untuk contoh ini, dan lewati Langkah 2.

2. Buat `metadata.rb` file dengan konten berikut, lalu simpan ke `apache2` direktori.

   ```
   name "apache2"
   version "0.1.0"
   ```

3. Di `apache2` direktori, buat `templates/default` direktori..

   > **Note**
   >
   > `templates/default`Direktori berfungsi untuk instance Amazon Linux, yang menggunakan `apache2.conf.erb` templat default. Instans Ubuntu 14.04 menggunakan `apache2.conf.erb` template khusus sistem operasi, yang ada di direktori. `templates/ubuntu-14.04` Jika Anda ingin kustomisasi diterapkan ke instance Ubuntu 14.04 juga, Anda harus mengganti template itu juga.

4. Salin `apache2.conf.erb`template bawaan ke `templates/default` direktori Anda. Buka file template, hapus komentar pada `ErrorDocument 500` baris, dan berikan pesan kesalahan khusus, sebagai berikut:

   ```
   ...
   ErrorDocument 500 "A custom error message."
   #ErrorDocument 404 /missing.html
   ...
   ```

5. Buat `.zip` arsip `opsworks_cookbooks` bernama`opsworks_cookbooks.zip`, lalu unggah file ke bucket Amazon Simple Storage Service (Amazon S3). Untuk kesederhanaan, buat arsip

publik. Rekam URL arsip untuk digunakan nanti. Anda juga dapat menyimpan buku masak Anda di arsip Amazon S3 pribadi atau dalam jenis repositori lainnya. Untuk informasi selengkapnya, lihat Repositori.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Saya Menghapus Bucket S3? .

> **Note**
>
> Untuk mempermudah, contoh ini menambahkan pesan kesalahan hardcode ke template. Untuk mengubahnya, Anda harus memodifikasi template dan menginstal ulang buku masak. Untuk memberi diri Anda fleksibilitas yang lebih besar, Anda dapat menentukan atribut kustom default untuk string kesalahan dalam file `customize.rb` atribut buku masak kustom dan menetapkan nilai atribut tersebut. `ErrorDocument 500` Misalnya, jika Anda memberi nama atribut`[:apache][:custom][:error500]`, baris yang sesuai `apache2.conf.erb` akan terlihat seperti berikut:
>
> ```
> ...
> ErrorDocument 500 <%= node[:apache][:custom][:error500] %>
> #ErrorDocument 404 /missing.html
> ...
> ```
>
> Anda kemudian dapat mengubah pesan kesalahan kustom kapan saja dengan mengganti`[:apache][:custom][:error500]`. Jika Anda menggunakan JSON khusus untuk mengganti atribut, Anda bahkan tidak perlu menyentuh buku masak.

Untuk menggunakan template kustom, buat tumpukan dan instal buku masak.

Untuk menggunakan template kustom

1. Buka konsol AWS OpsWorks Stacks, lalu pilih Add Stack.
2. Tentukan pengaturan standar berikut:
   - Nama - ApacheTemplate
   - Wilayah - AS Barat (Oregon)

- Kunci SSH default - Sebuah key pair Amazon Elastic Compute Cloud EC2 (Amazon)

  Jika Anda perlu membuat EC2 key pair Amazon, lihat Amazon EC2 Key Pairs. Perhatikan bahwa key pair harus memiliki wilayah AWS yang sama dengan instans.

Pilih Lanjutan>>, pilih Gunakan buku masak Chef kustom, untuk menentukan pengaturan berikut:

- Jenis repositori - Arsip Http
- URL Repositori — URL arsip buku masak yang Anda rekam sebelumnya

Terima nilai default untuk pengaturan lain, lalu pilih Tambah Tumpukan untuk membuat tumpukan.

3. Pilih Tambahkan lapisan, lalu tambahkan lapisan Java App Server ke tumpukan dengan pengaturan default.
4. Tambahkan instance 24/7 dengan pengaturan default ke layer, lalu mulai instance.

   Sebuah instance t2.micro sudah cukup untuk contoh ini.
5. Setelah instance online, hubungkan dengan SSH. `httpd.conf`File ada di `/etc/httpd/conf` direktori. File harus berisi `ErrorDocument` pengaturan kustom Anda, yang akan terlihat seperti berikut:

```
...
# Some examples:
ErrorDocument 500 "A custom error message."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
...
```

Load Balancing sebuah Layer

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks menyediakan dua opsi load balancing, Elastic Load Balancing HAProxydan, yang biasanya digunakan untuk menyeimbangkan beban di seluruh instance lapisan server aplikasi. Topik ini menjelaskan manfaat dan batasan masing-masing untuk membantu Anda memutuskan opsi mana yang akan dipilih saat menambahkan load balancing ke lapisan. Dalam beberapa kasus, pendekatan terbaik adalah menggunakan keduanya.

Pengakhiran SSL

HAProxy Lapisan bawaan tidak menangani penghentian SSL; Anda harus menghentikan SSL di server. Keuntungan dari pendekatan ini adalah lalu lintas dienkripsi hingga mencapai server. Namun, server harus menangani dekripsi, yang meningkatkan beban server. Selain itu, Anda harus meletakkan sertifikat SSL Anda di server aplikasi, yang lebih mudah diakses oleh pengguna.

Dengan Elastic Load Balancing, Anda dapat menghentikan SSL di load balancer. Ini mengurangi beban pada server aplikasi Anda, tetapi lalu lintas antara penyeimbang beban dan server tidak dienkripsi. Elastic Load Balancing juga memungkinkan Anda untuk menghentikan SSL di server, tetapi agak rumit untuk diatur.

Penskalaan

Jika lalu lintas masuk melebihi kapasitas penyeimbang HAProxy beban, Anda harus meningkatkan kapasitasnya secara manual.

Elastic Load Balancing secara otomatis menskalakan untuk menangani lalu lintas yang masuk. Untuk memastikan bahwa penyeimbang beban Elastic Load Balancing memiliki kapasitas yang cukup untuk menangani beban yang diharapkan saat pertama kali online, Anda dapat menghangatkannya terlebih dahulu.

Kegagalan Load Balancer

Jika instance hosting HAProxy server Anda gagal, itu bisa membuat seluruh situs Anda offline sampai Anda dapat memulai ulang instance.

Elastic Load Balancing lebih tahan kegagalan daripada. HAProxy Misalnya, ia menyediakan node load balancing di setiap Availability Zone yang memiliki EC2 instance terdaftar. Jika layanan di

satu zona terganggu, node lain terus menangani lalu lintas masuk. Untuk informasi selengkapnya, lihat Konsep Elastic Load Balancing.

Batas Waktu Idle

Kedua penyeimbang beban menghentikan koneksi jika server menganggur selama lebih dari nilai batas waktu idle yang ditentukan.

- HAProxy — Nilai batas waktu idle tidak memiliki batas atas.
- Elastic Load Balancing — Nilai batas waktu idle default adalah 60 detik, dengan maksimum 3600 detik (60 menit).

Batas waktu idle Elastic Load Balancing cukup untuk sebagian besar tujuan. Sebaiknya gunakan HAProxy jika Anda memerlukan batas waktu idle yang lebih lama. Misalnya:

- Koneksi HTTP yang berjalan lama yang digunakan untuk pemberitahuan push.
- Antarmuka administratif yang Anda gunakan untuk melakukan tugas yang bisa memakan waktu lebih dari 60 menit.

Pemetaan Berbasis URL

Anda mungkin ingin memiliki penyeimbang beban meneruskan permintaan masuk ke server tertentu berdasarkan URL permintaan. Misalnya, Anda memiliki sekelompok sepuluh server aplikasi yang mendukung aplikasi perdagangan online. Delapan server menangani katalog dan dua menangani pembayaran. Anda ingin mengarahkan semua permintaan HTTP terkait pembayaran ke server pembayaran, berdasarkan URL permintaan. Dalam hal ini, Anda akan mengarahkan semua URLs yang termasuk “pembayaran” atau “checkout” ke salah satu server pembayaran.

Dengan HAProxy, Anda dapat menggunakan pemetaan berbasis URL untuk mengarahkan URLs berisi string tertentu ke server tertentu. Untuk menggunakan pemetaan berbasis URL dengan AWS OpsWorks Stacks, Anda harus membuat file HAProxy konfigurasi khusus dengan mengganti `haproxy-default.erb` template di buku masak bawaan. `haproxy` Untuk informasi selengkapnya, lihat Manual HAProxy Konfigurasi danMenggunakan Template Kustom. Anda tidak dapat menggunakan pemetaan berbasis URL untuk permintaan HTTPS. Permintaan HTTPS dienkripsi, jadi tidak HAProxy ada cara untuk memeriksa URL permintaan.

Elastic Load Balancing memiliki dukungan terbatas untuk pemetaan URL. Untuk informasi selengkapnya, lihat Konfigurasi Pendengar untuk Elastic Load Balancing.

Rekomendasi: Sebaiknya gunakan Elastic Load Balancing untuk load balancing kecuali Anda memiliki persyaratan yang hanya dapat ditangani oleh. HAProxy Dalam hal ini, pendekatan terbaik mungkin menggabungkan keduanya dengan menggunakan Elastic Load Balancing sebagai penyeimbang beban front-end yang mendistribusikan lalu lintas masuk ke satu set server. HAProxy Untuk melakukannya:

- Siapkan HAProxy instance di setiap Availability Zone stack Anda untuk mendistribusikan permintaan ke server aplikasi zona tersebut.
- Tetapkan HAProxy instans ke penyeimbang beban Elastic Load Balancing, yang kemudian mendistribusikan permintaan masuk ke penyeimbang beban. HAProxy

Pendekatan ini memungkinkan Anda menggunakan HAProxy pemetaan berbasis URL untuk mendistribusikan berbagai jenis permintaan ke server aplikasi yang sesuai. Namun, jika salah satu HAProxy server offline, situs akan terus berfungsi karena load balancer Elastic Load Balancing secara otomatis mendistribusikan lalu lintas masuk ke server yang sehat. HAProxy Perhatikan bahwa Anda harus menggunakan Elastic Load Balancing sebagai penyeimbang beban front-end; HAProxy server tidak dapat mendistribusikan permintaan ke server lain. HAProxy

Migrasi dari Server Chef ke Stacks AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Karena AWS OpsWorks Stacks didasarkan pada Chef, migrasi dari Chef Server ke AWS OpsWorks Stacks relatif mudah. Topik ini memberikan panduan untuk memodifikasi kode Chef Server agar bekerja dengan AWS OpsWorks Stacks.

> **Note**
>
> Kami tidak menyarankan migrasi ke tumpukan menggunakan versi Chef lebih awal dari 11.10, yang didasarkan pada chef-solo dan tidak mendukung penelusuran atau tas data.

Topik

- Memetakan Peran ke Lapisan
- Menggunakan Tas Data
- Menggunakan Pencarian Chef
- Mengelola Buku Masak dan Resep
- Menggunakan Lingkungan Chef

Memetakan Peran ke Lapisan

Chef Server menggunakan peran untuk mewakili dan mengelola instance dengan tujuan dan konfigurasi yang sama, seperti serangkaian instance yang masing-masing meng-host server aplikasi Java. Lapisan AWS OpsWorks Stacks pada dasarnya memiliki tujuan yang sama dengan peran Chef. Lapisan adalah cetak biru untuk membuat satu set instance Amazon Elastic Compute Cloud (Amazon EC2) dengan konfigurasi yang sama, paket yang diinstal, prosedur penerapan aplikasi, dan sebagainya.

AWS OpsWorks Stacks mencakup satu set lapisan bawaan untuk beberapa jenis server aplikasi, penyeimbang HAProxy beban, master database MySQL, dan master pemantauan Ganglia. Misalnya, lapisan Java App Server bawaan adalah cetak biru untuk membuat instance yang meng-host server Tomcat.

Untuk bermigrasi ke AWS OpsWorks Stacks, Anda perlu mengaitkan setiap peran dengan lapisan yang menyediakan fungsionalitas yang setara. Untuk beberapa peran, Anda mungkin bisa menggunakan salah satu lapisan bawaan. Peran lain mungkin memerlukan berbagai tingkat penyesuaian. Mulailah dengan memeriksa fungsionalitas lapisan bawaan, termasuk resep yang terkait dengan masing-masing, untuk melihat apakah seseorang menyediakan setidaknya beberapa fungsi peran Anda. Untuk informasi selengkapnya tentang lapisan bawaan, lihat Lapisan danAWS OpsWorks Referensi Lapisan Tumpukan. Untuk memeriksa resep bawaan, lihat GitHubrepositori publik AWS OpsWorks Stacks.

Bagaimana Anda melanjutkan tergantung pada seberapa dekat Anda dapat mencocokkan lapisan untuk setiap peran, sebagai berikut.

Lapisan bawaan mendukung semua fungsionalitas peran

> Anda dapat menggunakan lapisan bawaan secara langsung, dengan penyesuaian kecil, jika perlu. Misalnya, jika peran mendukung server Tomcat, resep lapisan Java App Server mungkin

sudah menangani semua tugas peran, mungkin dengan beberapa penyesuaian sederhana. Misalnya, Anda dapat membuat resep bawaan layer menggunakan pengaturan konfigurasi Tomcat atau Apache kustom dengan mengganti atribut atau templat yang sesuai.

Lapisan bawaan mendukung beberapa, tetapi tidak semua, fungsionalitas peran

Anda mungkin dapat menggunakan lapisan bawaan dengan memperluas lapisan. Ini biasanya melibatkan penerapan resep khusus untuk mendukung fungsionalitas yang hilang dan menetapkan resep ke peristiwa siklus hidup lapisan. Misalnya, peran Anda menginstal server Redis pada instance yang sama yang meng-host server Tomcat. Anda dapat memperluas lapisan Java App Server agar sesuai dengan fungsionalitas peran dengan menerapkan resep khusus untuk menginstal Redis pada instance lapisan dan menetapkan resep ke acara Pengaturan lapisan.

Tidak ada lapisan bawaan yang cukup mendukung fungsionalitas peran

Menerapkan lapisan khusus. Misalnya, peran Anda mendukung server database MongoDB, yang tidak didukung oleh salah satu lapisan bawaan. Anda dapat memberikan dukungan tersebut dengan menerapkan resep untuk menginstal paket yang diperlukan, mengkonfigurasi server, dan sebagainya, dan menetapkan resep ke peristiwa siklus hidup lapisan kustom. Biasanya, Anda dapat menggunakan setidaknya beberapa resep peran untuk tujuan ini. Untuk informasi selengkapnya tentang cara menerapkan lapisan kustom, lihatMembuat Layer Server Tomcat Kustom.

Menggunakan Tas Data

Chef Server memungkinkan Anda untuk meneruskan data yang ditentukan pengguna ke resep Anda dengan menggunakan tas data.

- Anda menyimpan data dengan buku masak Anda, dan Chef menginstalnya di setiap instance.
- Anda dapat menggunakan tas data terenkripsi untuk data sensitif seperti kata sandi.

AWS OpsWorks Tumpukan mendukung tas data; resep dapat mengambil data menggunakan kode yang persis sama dengan Chef Server. Namun, dukungan memiliki batasan dan perbedaan berikut:

- Kantong data hanya didukung di Chef 11.10 Linux dan tumpukan yang lebih baru.

  Tumpukan Windows dan tumpukan Linux yang menjalankan versi Chef sebelumnya tidak mendukung kantong data.

- Anda tidak menyimpan tas data di repositori buku masak Anda.

  Sebagai gantinya, Anda menggunakan JSON khusus untuk mengelola data tas data Anda.

- AWS OpsWorks Stacks tidak mendukung kantong data terenkripsi.

  Jika Anda perlu menyimpan data sensitif dalam bentuk terenkripsi, seperti kata sandi atau sertifikat, sebaiknya simpan di bucket S3 pribadi. Anda kemudian dapat membuat resep khusus yang menggunakan Amazon SDK for Ruby untuk mengambil data. Sebagai contoh, lihat Menggunakan SDK untuk Ruby.

Untuk informasi selengkapnya, lihat Menggunakan Tas Data.

Menggunakan Pencarian Chef

Chef Server menyimpan informasi konfigurasi tumpukan, seperti alamat IP dan konfigurasi peran, di server. Resep menggunakan pencarian Chef untuk mengambil data ini. AWS OpsWorks Stacks menggunakan pendekatan yang agak berbeda. Misalnya, tumpukan Chef 11.10 Linux didasarkan pada mode lokal klien Chef, opsi klien Chef yang menjalankan versi ringan Chef Server (sering disebut Chef Zero) secara lokal pada instance. Chef Zero mendukung pencarian terhadap data yang disimpan dalam objek node instance.

Alih-alih menyimpan data tumpukan di server jarak jauh, AWS OpsWorks Stacks menambahkan satu set konfigurasi tumpukan dan atribut penerapan ke objek node setiap instance untuk setiap peristiwa siklus hidup. Atribut ini mewakili snapshot dari konfigurasi stack. Mereka menggunakan sintaks yang sama dengan Chef Server dan mewakili sebagian besar data yang perlu diambil resep dari server.

Anda sering tidak perlu memodifikasi kode yang bergantung pada pencarian resep Anda untuk Stacks. AWS OpsWorks Karena pencarian Chef beroperasi pada objek node, yang mencakup konfigurasi tumpukan dan atribut penerapan, kueri penelusuran di AWS OpsWorks Stacks biasanya bekerja persis seperti yang mereka lakukan dengan Chef Server.

Pengecualian utama disebabkan oleh fakta bahwa konfigurasi tumpukan dan atribut penerapan hanya berisi data yang diketahui AWS OpsWorks Stacks saat menginstal atribut pada instance. Jika Anda membuat atau memodifikasi atribut secara lokal pada instance tertentu, perubahan tersebut tidak menyebar kembali ke AWS OpsWorks Tumpukan dan tidak dimasukkan ke dalam konfigurasi tumpukan dan atribut penerapan yang diinstal pada instance lain. Anda dapat menggunakan pencarian untuk mengambil nilai atribut hanya pada instance itu. Untuk informasi selengkapnya, lihat Menggunakan Pencarian Chef.

Untuk kompatibilitas dengan Chef Server, AWS OpsWorks Stacks menambahkan satu set `role` atribut ke objek node, yang masing-masing berisi salah satu atribut layer stack. Jika resep Anda digunakan `roles` sebagai kunci pencarian, Anda tidak perlu mengubah kode pencarian. Kueri secara otomatis mengembalikan data untuk lapisan yang sesuai. Misalnya, kueri berikut mengembalikan atribut `php-app` layer.

```
phpserver = search(:node, "layers:php-app").first
```

```
phpserver = search(:node, "roles:php-app").first
```

Mengelola Buku Masak dan Resep

AWS OpsWorks Stacks dan Chef Server menangani buku masak dan resep agak berbeda. Dengan Server Chef:

- Anda menyediakan semua buku masak, baik dengan menerapkannya sendiri atau dengan menggunakan buku masak komunitas.
- Anda menyimpan buku masak di server.
- Anda menjalankan resep secara manual atau pada jadwal reguler.

Dengan AWS OpsWorks Tumpukan:

- AWS OpsWorks Stacks menyediakan satu atau lebih buku masak untuk masing-masing lapisan bawaan. Buku masak ini menangani tugas-tugas standar, seperti menginstal dan mengonfigurasi perangkat lunak lapisan bawaan dan menerapkan aplikasi.

  Untuk menangani tugas yang tidak dilakukan oleh buku masak bawaan, Anda menambahkan buku masak khusus ke tumpukan Anda atau menggunakan buku masak komunitas.
- Anda menyimpan buku masak AWS OpsWorks Stacks di repositori jarak jauh, seperti bucket S3 atau repositori Git.

  Untuk informasi selengkapnya, lihat Menyimpan Buku Masak.
- Anda dapat mengeksekusi resep secara manual, tetapi Anda biasanya memiliki AWS OpsWorks Stacks mengeksekusi resep untuk Anda sebagai respons terhadap serangkaian peristiwa siklus hidup yang terjadi pada titik-titik penting selama siklus hidup instans.

Untuk informasi selengkapnya, lihat Mengeksekusi Resep.

- AWS OpsWorks Stacks mendukung Berkshelf pada tumpukan Chef 11.10 saja. Jika Anda menggunakan Berkshelf untuk mengelola dependensi buku masak Anda, Anda tidak dapat menggunakan tumpukan yang menjalankan Chef 11.4 atau versi sebelumnya.

  Untuk informasi selengkapnya, lihat Menggunakan Berkshelf.

Topik

- Menyimpan Buku Masak
- Mengeksekusi Resep

Menyimpan Buku Masak

Dengan Chef Server, Anda menyimpan buku masak Anda di server dan menyebarkannya dari server ke instance. Dengan AWS OpsWorks Stacks, Anda menyimpan buku masak di repositori— arsip S3 atau HTTP atau repositori Git atau Subversion. Anda menentukan informasi yang dibutuhkan AWS OpsWorks Stacks untuk mengunduh kode dari repositori ke instance tumpukan saat Anda menginstal buku masak.

Untuk bermigrasi dari Chef Server, Anda harus meletakkan buku masak Anda di salah satu repositori ini. Untuk informasi tentang cara menyusun repositori buku masak, lihat. Repositori

Mengeksekusi Resep

Di AWS OpsWorks Stacks, setiap lapisan memiliki serangkaian peristiwa siklus hidup —Setup, Configure, Deploy, Undeploy, dan Shutdown—yang masing-masing terjadi pada titik kunci selama siklus hidup instance. Untuk menjalankan resep kustom, Anda biasanya menetapkannya ke acara yang sesuai pada layer yang sesuai. Ketika acara terjadi, AWS OpsWorks Stacks menjalankan resep terkait. Misalnya, peristiwa Setup terjadi setelah instance selesai booting, jadi Anda biasanya menetapkan resep untuk acara ini yang melakukan tugas-tugas seperti menginstal dan mengonfigurasi paket dan memulai layanan.

Anda dapat menjalankan resep secara manual dengan menggunakan perintah tumpukan Execute Recipes. Perintah ini berguna untuk pengembangan dan pengujian, tetapi Anda juga dapat menggunakannya untuk mengeksekusi resep yang tidak dipetakan ke peristiwa siklus hidup. Anda juga dapat menggunakan perintah Execute Recipes untuk memicu peristiwa Setup dan Configure secara manual.

Selain konsol AWS OpsWorks Stacks, Anda dapat menggunakan AWS CLI SDKsatau untuk mengeksekusi resep. Alat-alat ini mendukung semua tindakan AWS OpsWorks Stacks API, tetapi lebih mudah digunakan daripada API. Gunakan perintah CLI create-deployment untuk memicu peristiwa siklus hidup, yang menjalankan semua resep terkait. Anda juga dapat menggunakan perintah ini untuk menjalankan satu atau lebih resep tanpa memicu suatu peristiwa. Kode SDK yang setara tergantung pada bahasa tertentu, tetapi umumnya mirip dengan perintah CLI.

Contoh berikut menjelaskan dua cara untuk menggunakan perintah `create-deployment` CLI untuk mengotomatiskan penerapan aplikasi.

- Menerapkan aplikasi Anda pada jadwal reguler dengan menambahkan lapisan kustom dengan satu instance ke tumpukan Anda.

  Tambahkan resep Pengaturan kustom ke lapisan yang membuat `cron` pekerjaan pada instance untuk menjalankan perintah pada jadwal yang ditentukan. Untuk contoh cara menggunakan resep untuk membuat `cron` pekerjaan, lihatMenjalankan Cron Jobs di Instans Linux.
- Tambahkan tugas ke pipeline integrasi berkelanjutan Anda yang menggunakan perintah `create-deployment` CLI untuk menerapkan aplikasi.

Menggunakan Lingkungan Chef

AWS OpsWorks Tumpukan tidak mendukung lingkungan Chef; `node.chef_environment` selalu kembali`_default`.

AWS OpsWorks Referensi Lapisan Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setiap instance yang digunakan AWS OpsWorks Stacks harus menjadi anggota dari setidaknya satu lapisan, yang mendefinisikan peran instance dalam tumpukan dan mengontrol detail pengaturan dan konfigurasi instance, menginstal paket, menerapkan aplikasi, dan sebagainya. Untuk informasi

selengkapnya tentang cara menggunakan AWS OpsWorks Stacks untuk membuat dan mengelola lapisan, lihatLapisan.

Setiap deskripsi lapisan menyertakan daftar resep bawaan yang dijalankan AWS OpsWorks Stacks untuk setiap peristiwa siklus hidup lapisan. Resep-resep itu disimpan di https://github.com/aws/opsworks-cookbooks. Perhatikan bahwa daftar hanya menyertakan resep yang dijalankan langsung oleh AWS OpsWorks Stacks. Resep-resep itu terkadang menjalankan resep tergantung, yang tidak terdaftar. Untuk melihat daftar lengkap resep untuk acara tertentu, termasuk resep dependen dan khusus, periksa daftar lari di log Chef acara siklus hidup yang sesuai.

Topik

- HAProxy Referensi Lapisan
- HAProxy AWS OpsWorks Lapisan Tumpukan
- Referensi Lapisan MySQL
- Lapisan MySQL OpsWorks
- Referensi Lapisan Server Aplikasi
- Lapisan Server Aplikasi
- Referensi Lapisan Cluster ECS
- Referensi Lapisan Kustom
- Referensi Lapisan Lainnya
- Lapisan lainnya

HAProxy Referensi Lapisan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

HAProxy Lapisan menggunakan HAProxy—penyeimbang TCP/HTTP beban kinerja tinggi yang andal — untuk menyediakan penyeimbangan beban ketersediaan tinggi dan layanan proxy untuk aplikasi berbasis TCP dan HTTP. Ini sangat berguna untuk situs web yang harus dirayapi di bawah beban yang sangat tinggi sambil membutuhkan persistensi atau pemrosesan Layer 7.

HAProxy memantau lalu lintas dan menampilkan statistik dan kesehatan instance terkait pada halaman web. Secara default, URI adalah http://*DNSName*/haproxy? statistik, di *DNSName* mana nama DNS HAProxy instance.

Nama pendek: lb

Kompatibilitas: HAProxy Lapisan kompatibel dengan lapisan berikut: custom, db-master, dan memcached.

Port terbuka: HAProxy memungkinkan akses publik ke port 22 (SSH), 80 (HTTP), dan 443 (HTTPS).

Autoassign alamat IP Elastis: Aktif secara default

Volume EBS default: Tidak

Grup keamanan default: AWS-OpsWorks-LB-Server

Konfigurasi: Untuk mengkonfigurasi HAProxy lapisan, Anda harus menentukan yang berikut:

- Pemeriksaan Kesehatan URI (default: http://*DNSName*/).
- Statistik URI (default: http://*DNSName*/haproxy? statistik).
- Kata sandi statistik (opsional).
- Metode pemeriksaan kesehatan (opsional). Secara default, HAProxy menggunakan metode HTTP OPTIONS. Anda juga dapat menentukan GET atau HEAD.
- Aktifkan statistik (opsional)
- Pelabuhan. Secara default, AWS OpsWorks Stacks mengonfigurasi HAProxy untuk menangani lalu lintas HTTP dan HTTPS. Anda dapat mengonfigurasi HAProxy untuk menangani hanya satu atau yang lain dengan mengganti templat konfigurasi Chef,`haproxy.cfg.erb`.

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- haproksi

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version
- haproxy: :konfigurasi

Menyebarkan resep:

- menyebarkan: :default
- haproxy: :konfigurasi

Resep shutdown:

- opsworks_shutdown: :default
- haproxy: :berhenti

Instalasi:

- AWS OpsWorks Stacks menggunakan penginstal paket instance untuk menginstal HAProxy ke lokasi defaultnya.
- Anda harus mengatur syslog untuk mengarahkan file log ke lokasi tertentu. Untuk informasi selengkapnya, lihat HAProxy.

HAProxy AWS OpsWorks Lapisan Tumpukan

> **Note**
>
> Lapisan ini hanya tersedia untuk Chef 11 dan tumpukan berbasis Linux sebelumnya.

Lapisan AWS OpsWorks Stacks adalah HAProxy lapisan AWS OpsWorks Stacks yang menyediakan cetak biru untuk instance yang menampung server—keseimbangan beban kinerja tinggi yang HAProxyandal. TCP/HTTP Satu contoh kecil biasanya cukup untuk menangani semua lalu lintas server aplikasi.

> **Note**
>
> Tumpukan terbatas pada satu wilayah. Untuk mendistribusikan aplikasi Anda di beberapa wilayah, Anda harus membuat tumpukan terpisah untuk setiap wilayah.

Untuk membuat HAProxy layer

1. Di panel navigasi, klik Layers.
2. Pada halaman Layers, klik Add a Layer atau + Layer. Untuk tipe Layer, pilih HAProxy.

Lapisan ini memiliki pengaturan konfigurasi berikut, yang semuanya opsional.

HAProxy statistik

Apakah lapisan mengumpulkan dan menampilkan statistik. Nilai defaultnya adalah Ya.

URL Statistik

Jalur URL halaman statistik. URL lengkapnya adalah http://*DNSName*`StatisticsPath`, di *DNSName* mana nama DNS instance terkait. `StatisticsPath`Nilai defaultnya adalah / haproxy? statistik, yang sesuai dengan sesuatu seperti: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/haproxy?stats.

Nama pengguna statistik

Nama pengguna halaman statistik, yang harus Anda berikan untuk melihat halaman statistik. Nilai defaultnya adalah “opsworks”.

Kata sandi statistik

Kata sandi halaman statistik, yang harus Anda berikan untuk melihat halaman statistik. Nilai default adalah string yang dihasilkan secara acak.

URL pemeriksaan kesehatan

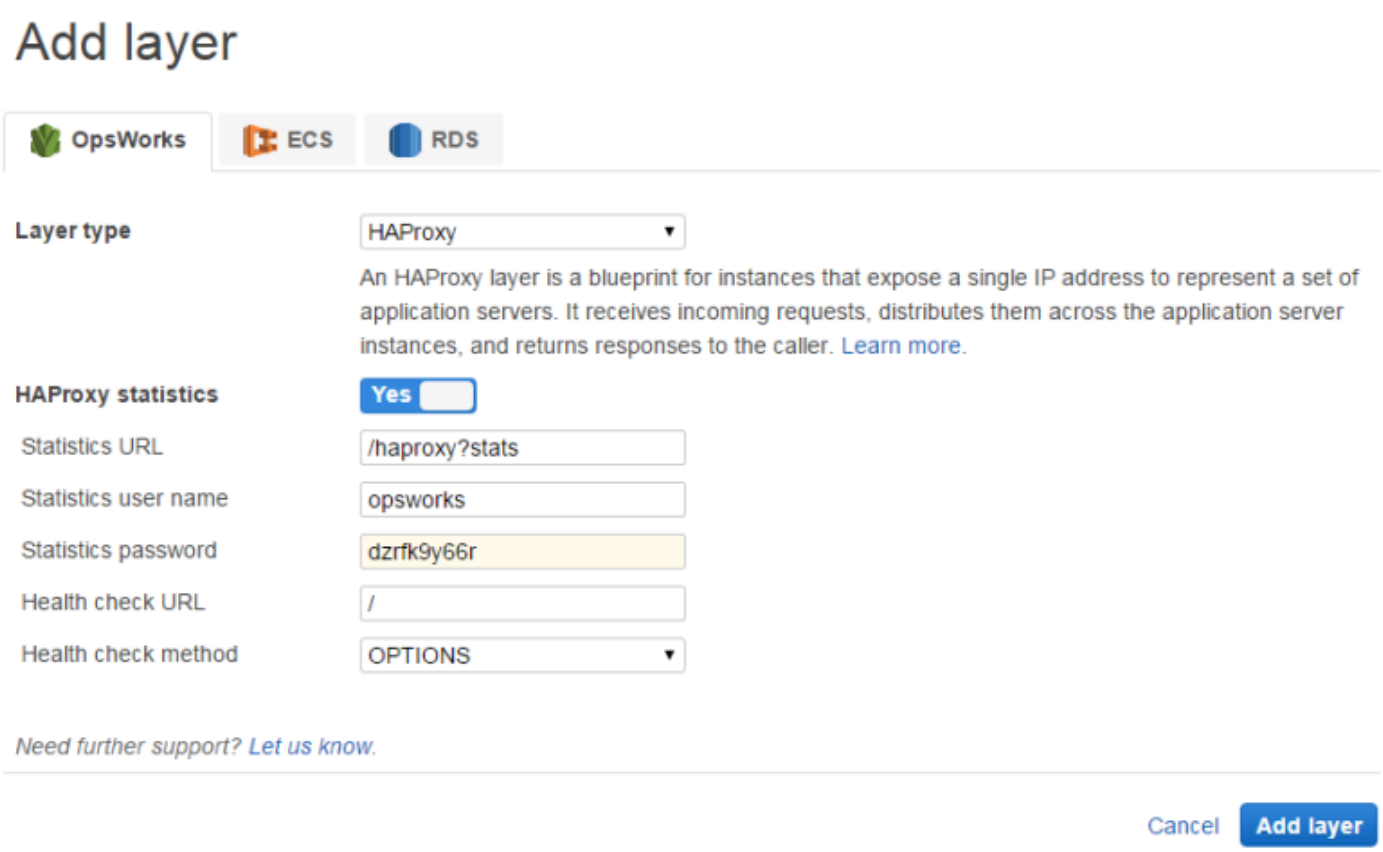
Sufiks URL pemeriksaan kesehatan. HAProxy menggunakan URL ini untuk secara berkala memanggil metode HTTP pada setiap instance server aplikasi untuk menentukan apakah instance berfungsi. Jika pemeriksaan kesehatan gagal, HAProxy hentikan perutean lalu lintas ke instance sampai dimulai ulang, baik secara manual atau melalui penyembuhan otomatis. Nilai default untuk akhiran URL adalah "/", yang sesuai dengan halaman beranda instance server: http://*DNSName*/.

Metode pemeriksaan Kesehatan

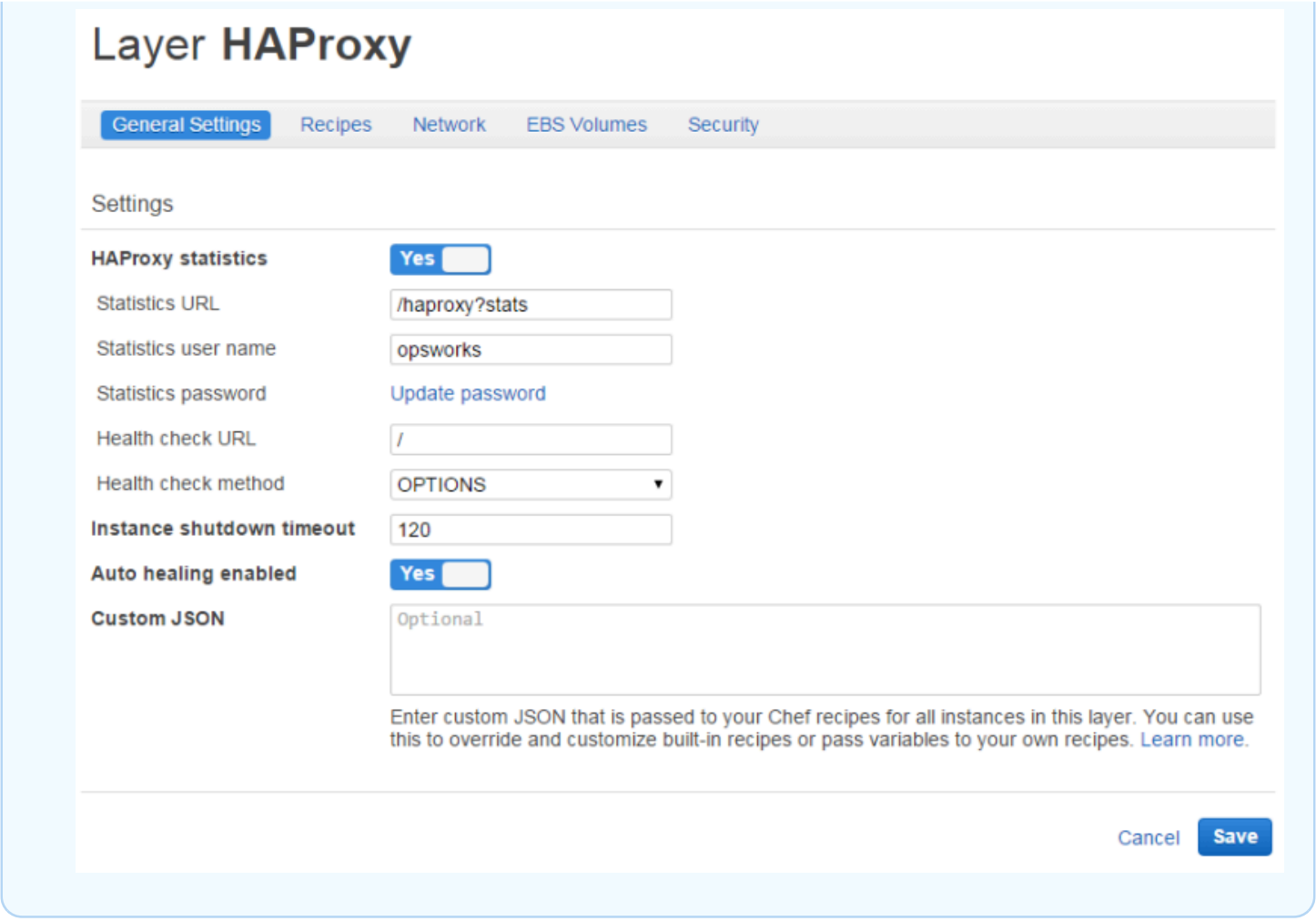
Metode HTTP yang akan digunakan untuk memeriksa apakah instance berfungsi. Nilai defaultnya adalah OPTIONS dan Anda juga dapat menentukan GET atau HEAD. Untuk informasi lebih lanjut, lihat httpchk.

Grup keamanan khusus

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Pastikan bahwa grup memiliki pengaturan yang benar untuk memungkinkan lalu lintas antar lapisan. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.



> **Note**
>
> Rekam kata sandi untuk digunakan nanti; AWS OpsWorks Tumpukan tidak memungkinkan Anda untuk melihat kata sandi setelah Anda membuat lapisan. Namun, Anda dapat memperbarui kata sandi dengan membuka halaman Edit lapisan dan mengklik Perbarui kata sandi pada tab Pengaturan Umum.



Bagaimana HAProxy Layer Bekerja

Secara default, HAProxy lakukan hal berikut:

- Mendengarkan permintaan pada port HTTP dan HTTPS.

  Anda dapat mengonfigurasi HAProxy untuk mendengarkan hanya pada port HTTP atau HTTPS dengan mengganti templat konfigurasi Chef,`haproxy.cfg.erb`.
- Rutekan lalu lintas masuk ke instance yang merupakan anggota lapisan server aplikasi apa pun.

  Secara default, AWS OpsWorks Stacks mengonfigurasi HAProxy untuk mendistribusikan lalu lintas ke instance yang merupakan anggota lapisan server aplikasi apa pun. Anda dapat, misalnya, memiliki tumpukan dengan lapisan Rails App Server dan PHP App Server, dan HAProxy master mendistribusikan lalu lintas ke instance di kedua lapisan. Anda dapat mengonfigurasi perutean default dengan menggunakan resep khusus.

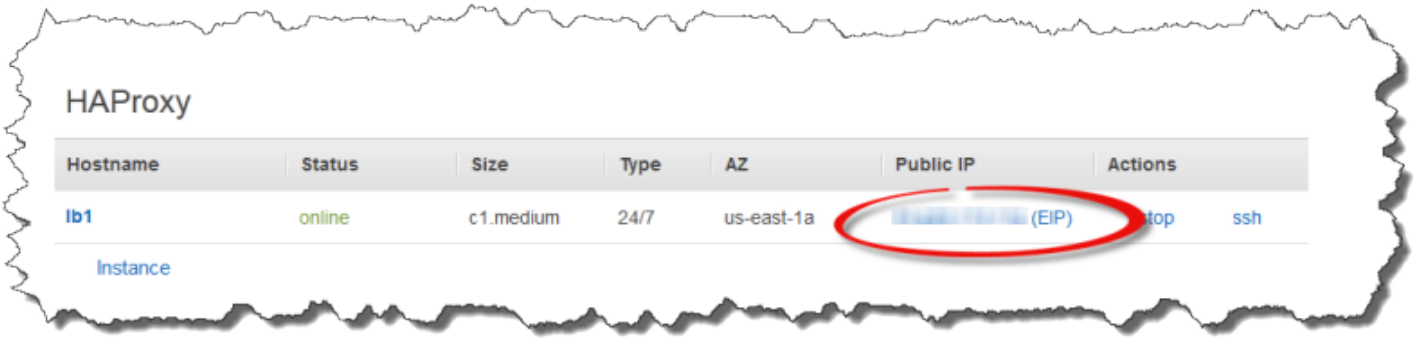
- Rutekan lalu lintas di beberapa Availability Zone.

  Jika satu Availability Zone turun, load balancer merutekan lalu lintas masuk ke instance di zona lain sehingga aplikasi Anda terus berjalan tanpa gangguan. Untuk alasan ini, praktik yang disarankan adalah mendistribusikan server aplikasi Anda di beberapa Availability Zone.
- Secara berkala menjalankan metode pemeriksaan kesehatan yang ditentukan pada setiap instance server aplikasi untuk menilai kesehatannya.

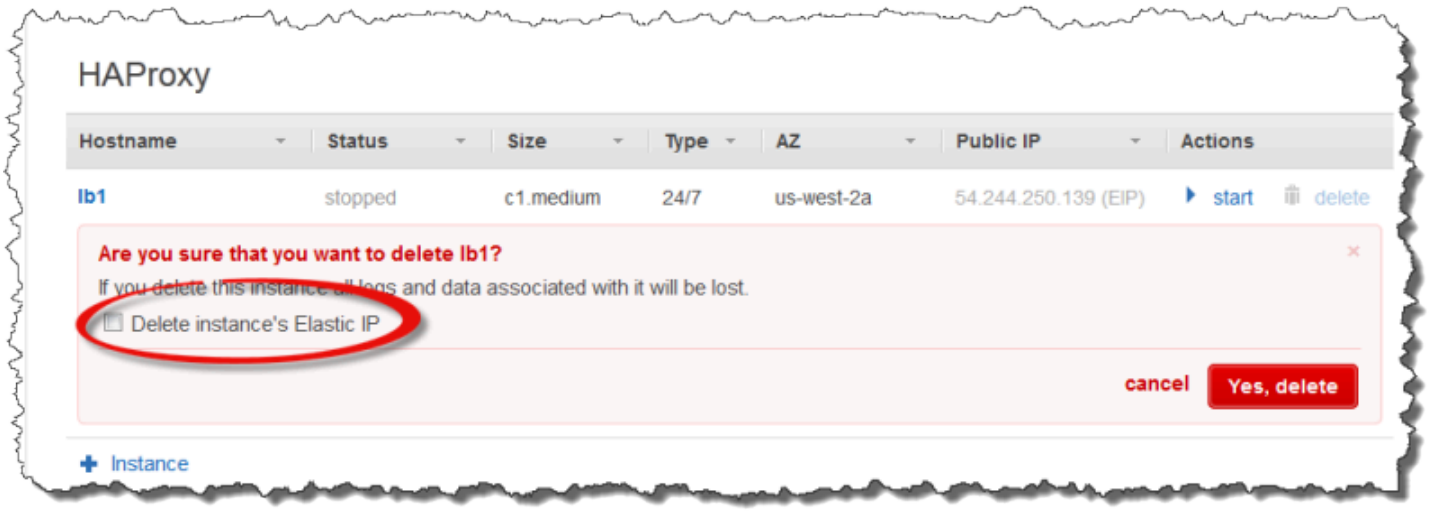
  Jika metode tidak kembali dalam periode batas waktu tertentu, instance dianggap gagal dan HAProxy menghentikan permintaan routing ke instance. AWS OpsWorks Stacks juga menyediakan cara untuk secara otomatis mengganti instance yang gagal. Untuk informasi selengkapnya, lihat Menggunakan Penyembuhan Otomatis. Anda dapat mengubah metode pemeriksaan kesehatan saat Anda membuat layer.
- Mengumpulkan statistik dan secara opsional menampilkannya di halaman web.

> **Important**
>
> Agar pemeriksaan kesehatan berfungsi dengan benar dengan metode OPTIONS default, aplikasi Anda harus mengembalikan kode status 2xx atau 3xx.

Secara default, ketika Anda menambahkan instance ke HAProxy lapisan, AWS OpsWorks Stacks memberinya alamat IP Elastis untuk mewakili aplikasi, yang bersifat publik ke dunia. Karena alamat IP Elastic HAProxy instance adalah satu-satunya URL aplikasi yang diekspos secara publik, Anda tidak perlu membuat dan mengelola nama domain publik untuk instance server aplikasi yang mendasarinya. Anda dapat memperoleh alamat dengan membuka halaman Instances dan memeriksa alamat IP publik instans, seperti yang ditunjukkan oleh ilustrasi berikut. Alamat yang diikuti oleh (EIP) adalah alamat IP elastis. Untuk informasi selengkapnya tentang alamat IP Elastis, lihat Alamat IP Elastis (EIP).



Saat Anda menghentikan sebuah HAProxy instance, AWS OpsWorks Stacks mempertahankan alamat IP Elastis dan menugaskannya kembali ke instance saat Anda memulai ulang. Jika Anda menghapus sebuah HAProxy instance, secara default, AWS OpsWorks Stacks menghapus alamat IP instans. Untuk mempertahankan alamat, hapus opsi Elastic IP instans Delete, seperti yang ditunjukkan pada ilustrasi berikut.



Opsi ini memengaruhi apa yang terjadi ketika Anda menambahkan instance baru ke layer untuk mengganti instance yang dihapus:

- Jika Anda mempertahankan alamat IP Elastic instans yang dihapus, AWS OpsWorks Stacks akan menetapkan alamat tersebut ke instans baru.
- Jika tidak, AWS OpsWorks Stacks menetapkan alamat IP Elastis baru ke instans dan Anda harus memperbarui pengaturan registrar DNS Anda untuk memetakan ke alamat baru.

Ketika instance server aplikasi online atau off line — baik secara manual atau sebagai konsekuensi dari penskalaan otomatis atau penyembuhan otomatis — konfigurasi penyeimbang beban harus diperbarui untuk mengarahkan lalu lintas ke rangkaian instance online saat ini. Tugas ini ditangani secara otomatis oleh resep bawaan lapisan:

- Saat instance baru muncul, AWS OpsWorks Stacks memicu peristiwa siklus hidup Konfigurasi. Resep Konfigurasi bawaan HAProxy lapisan memperbarui konfigurasi penyeimbang beban sehingga juga mendistribusikan permintaan ke instance server aplikasi baru.
- Ketika instance off line atau instance gagal dalam pemeriksaan kesehatan, AWS OpsWorks Stacks juga memicu peristiwa siklus hidup Konfigurasi. Resep HAProxy Konfigurasi memperbarui konfigurasi penyeimbang beban untuk merutekan lalu lintas ke hanya instance online yang tersisa.

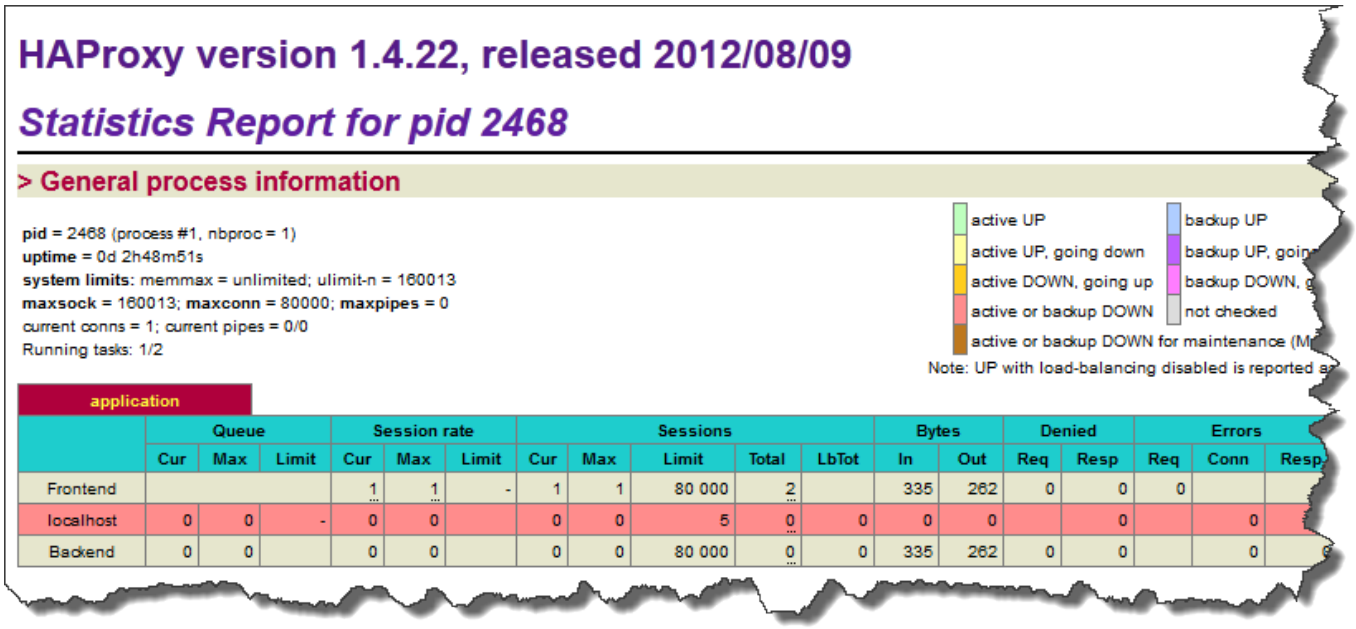
Terakhir, Anda juga dapat menggunakan domain khusus dengan HAProxy layer. Untuk informasi selengkapnya, lihat Menggunakan Domain Kustom.

Halaman Statistik

Jika Anda telah mengaktifkan halaman statistik, akan HAProxy menampilkan halaman yang berisi berbagai metrik di URL yang ditentukan.

Untuk melihat HAProxy statistik

1. Dapatkan nama DNS Publik HAProxy instans dari halaman Detail instans dan salin.
2. Pada halaman Layers, klik HAProxyuntuk membuka halaman detail layer.
3. Dapatkan URL statistik dari detail lapisan dan tambahkan ke nama DNS Publik. Misalnya:**`http://ec2-54-245-102-172.us-west-2.compute.amazonaws.com/haproxy?stats`**. untuk itu.
4. Tempel URL dari langkah sebelumnya ke browser Anda dan gunakan nama pengguna dan kata sandi yang Anda tentukan saat Anda membuat layer untuk membuka halaman statistik.



Referensi Lapisan MySQL

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan MySQL mendukung MySQL, sistem manajemen basis data relasional yang banyak digunakan. AWS OpsWorks Stacks menginstal versi terbaru yang tersedia, yang tergantung pada sistem operasi. Jika Anda menambahkan instance MySQL, informasi akses yang diperlukan disediakan untuk lapisan server aplikasi. Anda harus menulis resep Chef khusus untuk menyiapkan konfigurasi master—master atau master—slave.

Nama pendek: db-master

Kompatibilitas: Lapisan MySQL kompatibel dengan lapisan berikut: custom, lb, memcached, monitoring-master, nodejs-app, php-app, rails-app, dan web.

Port terbuka: Lapisan MySQL memungkinkan akses publik ke port 22 (SSH) dan semua port dari server web stack, server khusus, dan server aplikasi Rails, PHP, dan Node.js.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Ya, di `/vol/mysql`

Grup keamanan default: AWS-OpsWorks-DB-Master -Server

Konfigurasi: Untuk mengkonfigurasi lapisan MySQL, Anda harus menentukan yang berikut:

- Kata sandi pengguna root
- Mesin MySQL

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys

- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- mysql: :server
- dependensi
- menyebarkan: :mysql

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version
- menyebarkan: :mysql

Menyebarkan resep:

- menyebarkan: :default
- menyebarkan: :mysql

Resep shutdown:

- opsworks_shutdown: :default
- mysql: :berhenti

Instalasi:

- AWS OpsWorks Stacks menggunakan penginstal paket instance untuk menginstal MySQL dan file lognya ke lokasi defaultnya. Untuk informasi selengkapnya, lihat Dokumentasi MySQL.

Lapisan MySQL OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk Chef 11 dan tumpukan berbasis Linux sebelumnya.

Lapisan OpsWorks MySQL menyediakan cetak biru untuk EC2 instance Amazon yang berfungsi sebagai master database MySQL. Resep bawaan membuat database untuk setiap aplikasi yang telah dikerahkan ke lapisan server aplikasi. Misalnya, jika Anda menerapkan aplikasi PHP “myapp,” resep membuat database “myapp”.

Lapisan MySQL memiliki pengaturan konfigurasi berikut.

Kata sandi pengguna root MySQL

(Wajib) Kata sandi pengguna root.

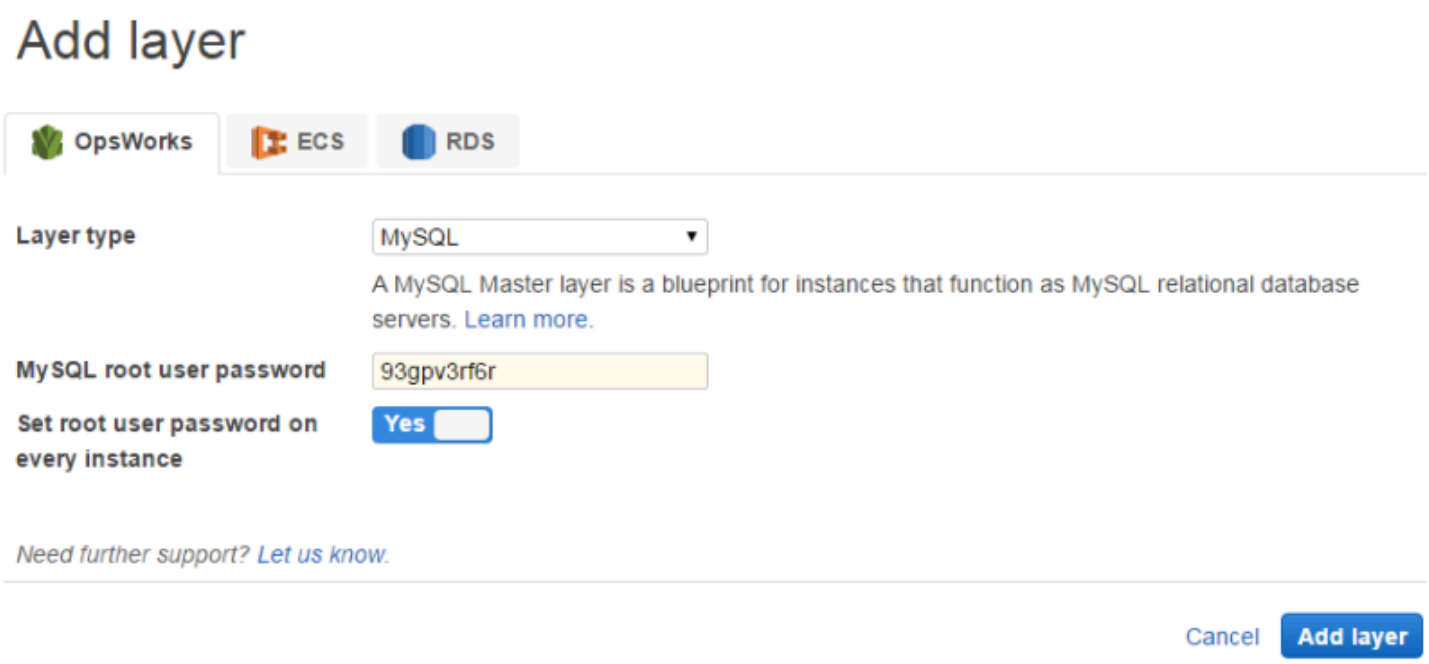
Tetapkan kata sandi pengguna root pada setiap contoh

(Opsional) Apakah kata sandi pengguna root disertakan dalam konfigurasi tumpukan dan atribut penerapan yang diinstal pada setiap instance di tumpukan. Pengaturan defaultnya adalah Ya.

Jika Anda menetapkan nilai ini ke Tidak, AWS OpsWorks Stacks akan meneruskan kata sandi root hanya ke instance server aplikasi.

Grup keamanan khusus

(Opsional) Grup keamanan khusus yang akan dikaitkan dengan lapisan. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.



Anda dapat menambahkan satu atau lebih instance ke layer, yang masing-masing mewakili master database MySQL yang terpisah. Anda kemudian dapat melampirkan instance ke aplikasi, yang menginstal informasi koneksi yang diperlukan di server aplikasi. Aplikasi kemudian dapat menggunakan informasi koneksi untuk terhubung ke server database instance.

Referensi Lapisan Server Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks mendukung beberapa aplikasi yang berbeda dan server halaman web statis.

Topik

- Referensi Lapisan AWS Flow (Ruby)
- Referensi Lapisan Server Aplikasi Java

- Referensi Lapisan Server Aplikasi Node.js
- Referensi Lapisan Server Aplikasi PHP
- Referensi Lapisan Server Aplikasi Rails
- Referensi Lapisan Server Web Statis

Referensi Lapisan AWS Flow (Ruby)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan AWS Flow (Ruby) menyediakan cetak biru untuk instance yang menampung aktivitas Amazon Simple Workflow Service dan pekerja alur kerja.

Nama pendek: aws-flow-ruby

Kompatibilitas: Lapisan AWS Flow (Ruby) kompatibel dengan PHP App Server, MySQL, Memcached, Ganglia, dan lapisan kustom.

Buka port: Tidak ada.

Peran IAM: aws-opsworks-ec 2- role-with-swf adalah peran AWS Flow (Ruby) standar yang dibuat AWS OpsWorks Stacks untuk Anda, jika diminta.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS Default: Tidak

Grup keamanan default: AWS-OpsWorks-AWS-Flow -Ruby-Server

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- opsworks_aws_flow_ruby: :pengaturan

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- mysql: :klien
- agent_version
- opsworks_aws_flow_ruby: :konfigurasi

Menyebarkan resep:

- menyebarkan: :default
- menyebarkan:: aws-flow-ruby

Undeploy resep:

- menyebarkan:: aws-flow-ruby-undeploy

Resep shutdown:

- opsworks_shutdown: :default

Referensi Lapisan Server Aplikasi Java

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan Java App Server mendukung server aplikasi Apache Tomcat 7.0.

Nama pendek: java-app

Kompatibilitas: Lapisan Java App Server kompatibel dengan lapisan berikut: custom, db-master, dan memcached.

Port terbuka: Lapisan Java App Server memungkinkan akses publik ke port 22 (SSH), 80 (HTTP), 443 (HTTPS), dan semua port dari penyeimbang beban.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS Default: Tidak

Grup keamanan default: AWS-OpsWorks-Java-App -Server

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs

- opsworks_ganglia: :klien
- opsworks_java: :pengaturan

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version
- opsworks_java: :konfigurasi

Menyebarkan resep:

- menyebarkan: :default
- menyebarkan: :java

Undeploy resep:

- menyebarkan: :java-undeploy

Resep shutdown:

- opsworks_shutdown: :default
- menyebarkan: :java-stop

Instalasi:

- Tomcat menginstal ke. `/usr/share/tomcat7`
- Untuk informasi selengkapnya tentang cara menghasilkan file log, lihat Masuk Tomcat.

Referensi Lapisan Server Aplikasi Node.js

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan Server Aplikasi Node.js mendukung server aplikasi Node.js, yang merupakan platform untuk mengimplementasikan server aplikasi jaringan yang sangat skalabel. Program ditulis dalam JavaScript, menggunakan asinkron berbasis peristiwa I/O untuk meminimalkan overhead dan memaksimalkan skalabilitas.

Nama pendek: nodejs-app

Kompatibilitas: Lapisan Server Aplikasi Node.js kompatibel dengan lapisan berikut: custom, db-master, memcached, dan monitoring-master.

Port terbuka: Lapisan Server Aplikasi Node.js memungkinkan akses publik ke port 22 (SSH), 80 (HTTP), 443 (HTTPS), dan semua port dari penyeimbang beban.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Tidak

Grup keamanan default: AWS-OpsWorks-nodejs-App -Server

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs

- opsworks_ganglia: :klien
- opsworks_nodejs
- opsworks_nodejs: :npm

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version
- opsworks_nodejs: :konfigurasi

Menyebarkan resep:

- menyebarkan: :default
- opsworks_nodejs
- opsworks_nodejs: :npm
- menyebarkan: :nodejs

Undeploy resep:

- menyebarkan: :nodejs-undeploy

Resep shutdown:

- opsworks_shutdown: :default
- menyebarkan: :nodejs-stop

Instalasi:

- Node.js menginstal ke`/usr/local/bin/node`.
- Untuk informasi selengkapnya tentang cara menghasilkan file log, lihat Cara masuk node.js di situs web Nodejitsu.

Konfigurasi aplikasi Node.js:

- File utama yang dijalankan oleh Node.js harus diberi nama `server.js` dan berada di direktori root aplikasi yang digunakan.
- Aplikasi Node.js harus diatur untuk mendengarkan pada port 80 (atau port 443, jika ada).

> **Note**
>
> Aplikasi Node.js yang menjalankan Express biasanya menggunakan kode berikut untuk mengatur port mendengarkan, di mana `process.env.PORT` mewakili port default dan menyelesaikan ke 80:
>
> ```
> app.set('port', process.env.PORT || 3000);
> ```
>
> Dengan AWS OpsWorks Stacks, Anda harus secara eksplisit menentukan port 80, sebagai berikut:
>
> ```
> app.set('port', 80);
> ```

Referensi Lapisan Server Aplikasi PHP

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan PHP App Server mendukung server aplikasi PHP dengan menggunakan Apache2 dengan mod_php.

Nama pendek: php-app

Kompatibilitas: Lapisan PHP App Server kompatibel dengan lapisan berikut: custom, db-master, memcached, monitoring-master, dan rails-app.

Port terbuka: Lapisan PHP App Server memungkinkan akses publik ke port 22 (SSH), 80 (HTTP), 443 (HTTPS), dan semua port dari penyeimbang beban.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Tidak

Grup keamanan default: AWS-OpsWorks-PHP-App -Server

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- mysql: :klien
- dependensi
- mod_php5_apache2

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version
- mod_php5_apache2: :php
- php: :mengkonfigurasi

Menyebarkan resep:

- menyebarkan: :default
- menyebarkan: :php

Undeploy resep:

- menyebarkan: :php-undeploy

Resep shutdown:

- opsworks_shutdown: :default
- apache2: :berhenti

Instalasi:

- AWS OpsWorks Stacks menggunakan penginstal paket instance untuk menginstal Apache2, mod_php dan file log terkait ke lokasi defaultnya. Untuk informasi lebih lanjut tentang instalasi, lihat Apache. Untuk informasi selengkapnya tentang pencatatan, lihat File Log.

Referensi Lapisan Server Aplikasi Rails

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan Rails App Server mendukung server aplikasi Ruby on Rails.

Nama pendek: rails-app

Kompatibilitas: Lapisan Rails App Server kompatibel dengan lapisan berikut: custom, db-master, memcached, monitoring-master, php-app.

Port: Lapisan Rails App Server memungkinkan akses publik ke port 22 (SSH), 80 (HTTP), 443 (HTTPS), dan semua port dari penyeimbang beban.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Tidak

Grup keamanan default: AWS-OpsWorks-Rails-App -Server

Konfigurasi: Untuk mengkonfigurasi layer Rails App Server, Anda harus menentukan yang berikut:

- Versi Ruby
- Tumpukan rel
- Versi Rubygems
- Apakah akan menginstal dan mengelola Bundler
- Versi Bundler

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- apache2 apache2: :mod_deflate
- penumpang_apache2
- penumpang_apache2: :mod_rails
- penumpang_apache2: :rel

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version
- rel: :konfigurasi

Menyebarkan resep:

- menyebarkan: :default
- menyebarkan: :rel

Undeploy resep:

- menyebarkan: :rails-undeploy

Resep shutdown:

- opsworks_shutdown: :default
- apache2: :berhenti

Instalasi:

- AWS OpsWorks Stacks menggunakan penginstal paket instance untuk menginstal Apache2 dengan mod_passenger, mod_rails, dan file log terkait ke lokasi defaultnya. Untuk informasi lebih lanjut tentang pemasangan, lihat Penumpang Phusion. Untuk informasi selengkapnya tentang pencatatan, lihat File Log.

Referensi Lapisan Server Web Statis

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan Web Server Statis menyajikan halaman HTML statis, yang dapat mencakup kode sisi klien seperti. JavaScript Hal ini didasarkan pada Nginx, yang merupakan open source HTTP, reverse proxy, dan mail proxy server.

Nama pendek: web

Kompatibilitas: Lapisan Server Web Statis kompatibel dengan lapisan berikut: kustom, db-master, memcached.

Port terbuka: Lapisan Server Web Statis memungkinkan akses publik ke port 22 (SSH), 80 (HTTP), 443 (HTTPS), dan semua port dari penyeimbang beban.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Tidak

Grup keamanan default: AWS-OpsWorks-Web-Server

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- nginx

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users

- agent_version

Menyebarkan resep:

- menyebarkan: :default
- menyebarkan: :web

Undeploy resep:

- menyebarkan: :web-undeploy

Resep shutdown:

- opsworks_shutdown: :default
- nginx: :berhenti

Instalasi:

- Nginx menginstal ke. `/usr/sbin/nginx`
- File log Nginx ada di. `/var/log/nginx`

Lapisan Server Aplikasi

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Lapisan ini hanya tersedia untuk Chef 11 dan tumpukan berbasis Linux sebelumnya.

AWS OpsWorks Stacks mendukung beberapa server aplikasi yang berbeda, di mana “aplikasi” menyertakan halaman web statis. Setiap jenis server memiliki lapisan AWS OpsWorks Stacks terpisah, dengan resep bawaan yang menangani pemasangan server aplikasi dan paket terkait apa pun pada setiap instance lapisan, penerapan aplikasi, dan sebagainya. Misalnya, lapisan Java App Server menginstal beberapa paket—termasuk Apache, Tomcat, dan OpenJDK—dan menyebarkan aplikasi Java ke setiap instance lapisan.

Berikut ini adalah prosedur dasar untuk menggunakan lapisan server aplikasi:

1. Buat salah satu jenis layer App Server yang tersedia.
2. Tambahkan satu atau lebih instance ke layer.
3. Buat aplikasi dan terapkan ke instance. Untuk informasi selengkapnya, lihat Aplikasi.
4. (Opsional) Jika layer memiliki beberapa instance, Anda dapat menambahkan penyeimbang beban, yang mendistribusikan lalu lintas masuk di seluruh instance. Untuk informasi selengkapnya, lihat HAProxy AWS OpsWorks Lapisan Tumpukan.

Topik

- Lapisan AWS Flow (Ruby)
- Lapisan AWS OpsWorks Tumpukan Server Aplikasi Java
- Lapisan AWS OpsWorks Tumpukan Server Aplikasi Node.js
- Lapisan AWS OpsWorks Tumpukan Server Aplikasi PHP
- Lapisan AWS OpsWorks Tumpukan Server Aplikasi Rails
- Lapisan AWS OpsWorks Tumpukan Server Web Statis

Lapisan AWS Flow (Ruby)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan AWS Flow (Ruby) adalah lapisan AWS OpsWorks Stacks yang menyediakan cetak biru untuk instance yang menampung aktivitas SWF Amazon dan pekerja alur kerja. Para pekerja diimplementasikan dengan menggunakan AWS Flow Framework for Ruby, yang merupakan kerangka kerja pemrograman yang menyederhanakan proses penerapan aplikasi asinkron terdistribusi sambil memberikan semua manfaat Amazon SWF. Ini sangat ideal untuk menerapkan aplikasi untuk mengatasi berbagai skenario, termasuk proses bisnis, pengkodean media, tugas yang berjalan lama, dan pemrosesan latar belakang.

Lapisan AWS Flow (Ruby) mencakup pengaturan konfigurasi berikut.

RubyGems versi

Versi Permata kerangka kerja.

Versi Bundler

Versi Bundler.

EC2 Profil contoh

Profil EC2 instans Amazon yang ditentukan pengguna untuk digunakan oleh instance lapisan. Profil ini harus memberikan izin untuk aplikasi yang berjalan pada instance lapisan untuk mengakses Amazon SWF.

Jika akun Anda tidak memiliki profil yang sesuai, Anda dapat memilih Profil baru dengan akses SWF agar AWS OpsWorks Stacks memperbarui profil atau Anda dapat memperbaruinya sendiri dengan menggunakan konsol IAM. Anda kemudian dapat menggunakan profil yang diperbarui untuk semua lapisan AWS Flow berikutnya. Berikut ini adalah deskripsi singkat tentang cara membuat profil dengan menggunakan konsol IAM. Untuk informasi selengkapnya, lihat Identity and Access Management di Amazon Simple Workflow Service.

Membuat profil untuk instans AWS Flow (Ruby)

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pilih Kebijakan di panel navigasi dan pilih Buat kebijakan untuk membuat kebijakan baru yang dikelola pelanggan.

3. Untuk Layanan, pilih SWF.
4. Untuk Tindakan, pilih Semua tindakan SWF (swf: *).
5. Untuk Nama Sumber Daya Amazon (ARN), masukkan ARN yang menentukan domain Amazon SWF mana yang dapat diakses pekerja. Pilih **`All resources`** untuk menyediakan akses ke semua domain.
6. Pilih Berikutnya.
7. Secara opsional, masukkan tag untuk mengidentifikasi kebijakan.
8. Pilih Berikutnya.
9. Setelah selesai, pilih Buat kebijakan.
10. Pilih Peran di panel navigasi dan pilih Buat peran.
11. Tentukan nama peran dan pilih Langkah Berikutnya. Anda tidak dapat mengubah nama setelah peran dibuat.
12. Pilih layanan AWS lalu pilih EC2.
13. Pilih Berikutnya.
14. Dari daftar Kebijakan izin, pilih kebijakan yang Anda buat sebelumnya.
15. Pilih Berikutnya.
16. Masukkan nama peran dan pilih Buat peran. Anda tidak dapat mengubah nama setelah peran dibuat.
17. Tentukan profil ini saat Anda membuat layer AWS Flow (Ruby) di AWS OpsWorks Stacks.

Lapisan AWS OpsWorks Tumpukan Server Aplikasi Java

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan Java App Server adalah lapisan AWS OpsWorks Stacks yang menyediakan cetak biru untuk instance yang berfungsi sebagai server aplikasi Java. Lapisan ini didasarkan pada Apache Tomcat 7.0 dan Open JDK 7. AWS OpsWorks Stacks juga menginstal pustaka konektor Java, yang memungkinkan aplikasi Java menggunakan `DataSource` objek JDBC untuk terhubung ke penyimpanan data back end.

Instalasi: Tomcat dipasang di`/usr/share/tomcat7`.

Halaman Add Layer menyediakan opsi konfigurasi berikut:

Opsi Java VM

Anda dapat menggunakan pengaturan ini untuk menentukan opsi VM Java kustom; tidak ada opsi default. Misalnya, satu set opsi umum adalah`-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. Jika Anda menggunakan Java VM Options, pastikan Anda melewati serangkaian opsi yang valid; AWS OpsWorks Tumpukan tidak memvalidasi string. Jika Anda mencoba meneruskan opsi yang tidak valid, server Tomcat biasanya gagal memulai, yang menyebabkan penyiapan gagal. Jika itu terjadi, Anda dapat memeriksa log Chef penyiapan instans untuk detailnya. Untuk informasi selengkapnya tentang cara melihat dan menafsirkan log Chef, lihatLog Koki.

Grup keamanan kustom

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Elastic Load Balancer

Anda dapat memasang penyeimbang beban Elastic Load Balancing ke instance layer. Untuk informasi selengkapnya, lihat Lapisan Elastic Load Balancing.

Anda dapat menentukan pengaturan konfigurasi lainnya dengan menggunakan JSON kustom atau file atribut kustom. Untuk informasi selengkapnya, lihat Konfigurasi Kustom.

> **Important**
>
> Jika aplikasi Java Anda menggunakan SSL, kami sarankan Anda menonaktifkan SSLv3 jika mungkin untuk mengatasi kerentanan yang dijelaskan dalam CVE-2014-3566. Untuk informasi selengkapnya, lihat Menonaktifkan Server SSLv3 Apache.

Topik

- Menonaktifkan Server SSLv3 Apache
- Konfigurasi Kustom
- Menerapkan Aplikasi Java

Menonaktifkan Server SSLv3 Apache

Untuk menonaktifkan SSLv3, Anda harus mengubah `SSLProtocol` pengaturan `ssl.conf` file server Apache. Untuk melakukannya, Anda harus mengganti file `ssl.conf.erb` template buku masak apache2 bawaan, yang digunakan oleh resep Pengaturan lapisan Java App Server untuk membuat. `ssl.conf` Detailnya tergantung pada sistem operasi mana yang Anda tentukan untuk instance lapisan. Berikut ini merangkum modifikasi yang diperlukan untuk sistem Amazon Linux dan Ubuntu. SSLv3 secara otomatis dinonaktifkan untuk sistem Red Hat Enterprise Linux (RHEL). Untuk informasi selengkapnya tentang cara mengganti template bawaan, lihatMenggunakan Template Kustom.

Amazon Linux

`ssl.conf.erb`File untuk sistem operasi ini ada di `apache2/templates/default/mods` direktori `apache2` buku masak. Berikut ini menunjukkan bagian yang relevan dari file bawaan.

```
...
#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>
```

Ganti `ssl.conf.erb` dan ubah `SSLProtocol` pengaturan sebagai berikut.

```
...
#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

`ssl.conf.erb`File untuk sistem operasi ini ada di `apache2/templates/ubuntu-14.04/mods` direktori `apache2` buku masak. Berikut ini menunjukkan bagian yang relevan dari file bawaan.

```
...
# The protocols to enable.
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all
...
```

Ubah pengaturan ini menjadi yang berikut.

```
...
# The protocols to enable.
# Available values: all, SSLv3, TLSv1.2
# SSL v2 is no longer supported
SSLProtocol all -SSLv3 -SSLv2
...
```

Konfigurasi Kustom

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks memperlihatkan pengaturan konfigurasi tambahan sebagai atribut bawaan, yang semuanya ada di namespace. `opsworks_java` Anda dapat menggunakan JSON kustom atau file atribut khusus untuk mengganti atribut bawaan dan menentukan nilai kustom. Misalnya, versi JVM dan Tomcat diwakili oleh built-in `jvm_version` dan `java_app_server_version` atribut, yang keduanya diatur ke 7. Anda dapat menggunakan JSON kustom atau file atribut kustom untuk mengatur salah satu atau keduanya ke 6. Contoh berikut menggunakan JSON kustom untuk mengatur kedua atribut ke 6:

```
{
  "opsworks_java": {
    "jvm_version": 6,
    "java_app_server_version" : 6
  }
}
```

Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom.

Contoh lain dari konfigurasi kustom adalah menginstal JDK kustom dengan mengganti atribut`use_custom_pkg_location`,`custom_pkg_location_url_debian`, dan. `custom_pkg_location_url_rhel`

Note

Jika Anda mengganti buku masak bawaan, Anda perlu memperbarui sendiri komponen-komponen itu.

Untuk informasi selengkapnya tentang atribut dan cara menggantinya, lihatMengesampingkan Atribut. Untuk daftar atribut bawaan, lihatopsworks_java Atribut.

Menerapkan Aplikasi Java

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik berikut menjelaskan cara menerapkan aplikasi ke instance lapisan Java App Server. Contohnya adalah untuk aplikasi JSP, tetapi pada dasarnya Anda dapat menggunakan prosedur yang sama untuk menginstal jenis aplikasi Java lainnya.

Anda dapat menyebarkan halaman JSP dari salah satu repositori yang didukung. Jika Anda ingin menyebarkan file WAR, perhatikan bahwa AWS OpsWorks Stacks secara otomatis mengekstrak file WAR yang disebarkan dari arsip Amazon S3 atau HTTP, tetapi tidak dari repositori Git atau Subversion. Jika Anda ingin menggunakan Git atau Subversion untuk file WAR, Anda dapat melakukan salah satu hal berikut:

- Simpan arsip yang diekstraksi di repositori.
- Simpan file WAR di repositori dan gunakan hook penyebaran Chef untuk mengekstrak arsip, seperti yang dijelaskan dalam contoh berikut.

Anda dapat menggunakan kait penerapan Chef untuk menjalankan aplikasi Ruby yang disediakan pengguna pada instance di salah satu dari empat tahap penerapan. Nama aplikasi menentukan tahapan. Berikut ini adalah contoh aplikasi Ruby bernama`before_migrate.rb`, yang mengekstrak file WAR yang telah digunakan dari repositori Git atau Subversion. Nama tersebut mengaitkan aplikasi dengan hook penerapan Checkout sehingga berjalan di awal operasi penyebaran, setelah kode diperiksa tetapi sebelum migrasi. Untuk informasi selengkapnya tentang cara menggunakan contoh ini, lihatMenggunakan Chef Deployment Hooks.

```
::Dir.glob(::File.join(release_path, '*.war')) do |archive_file|
  execute "unzip_#{archive_file}" do
    command "unzip #{archive_file}"
    cwd release_path
```

```
  end
end
```

> **Note**
>
> Saat Anda menerapkan pembaruan ke aplikasi JSP, Tomcat mungkin tidak mengenali pembaruan dan sebagai gantinya terus menjalankan versi aplikasi yang ada. Hal ini dapat terjadi, misalnya, jika Anda menerapkan aplikasi Anda sebagai file.zip yang hanya berisi halaman JSP. Untuk memastikan bahwa Tomcat menjalankan versi yang paling baru digunakan, direktori root proyek harus menyertakan direktori WEB-INF yang berisi file. `web.xml` `web.xml`File dapat berisi berbagai konten, tetapi berikut ini cukup untuk memastikan bahwa Tomcat mengenali pembaruan dan menjalankan versi aplikasi yang saat ini digunakan. Anda tidak perlu mengubah versi untuk setiap pembaruan. Tomcat akan mengenali pembaruan meskipun versinya tidak berubah.
>
> ```
> <context-param>
>   <param-name>appVersion</param-name>
>   <param-value>0.1</param-value>
> </context-param>
> ```

Topik

- Menerapkan Aplikasi JSP
- Menyebarkan Aplikasi JSP dengan Back-End Database

Menerapkan Aplikasi JSP

Untuk menerapkan aplikasi JSP, tentukan nama dan informasi repositori. Anda juga dapat menentukan domain dan pengaturan SSL secara opsional. Untuk informasi selengkapnya tentang cara membuat aplikasi, lihatMenambahkan Aplikasi. Prosedur berikut menunjukkan cara membuat dan menyebarkan halaman JSP sederhana dari arsip Amazon S3 publik. Untuk informasi tentang cara menggunakan jenis repositori lain, termasuk arsip Amazon S3 pribadi, lihat. Sumber Aplikasi

Contoh berikut menunjukkan halaman JSP, yang hanya menampilkan beberapa informasi sistem.

```
<%@ 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**
>
> Prosedur berikut mengasumsikan bahwa Anda sudah terbiasa dengan dasar-dasar membuat tumpukan, menambahkan instance ke lapisan, dan sebagainya. Jika Anda baru mengenal AWS OpsWorks Stacks, Anda harus melihatnya Memulai dengan Chef 11 Linux Stacks terlebih dahulu.

Untuk menyebarkan halaman JSP dari arsip Amazon S3

1. Buat tumpukan dengan lapisan Java App Server, tambahkan instance 24/7 ke layer, dan mulai.
2. Salin kode ke file bernama`simplejsp.jsp`, letakkan file di folder bernama`simplejsp`, dan buat `.zip` arsip folder. Namanya sewenang-wenang; Anda dapat menggunakan nama file atau folder apa pun yang Anda inginkan. Anda juga dapat menggunakan jenis arsip lain, termasuk file gzip, bzip2, tarball, atau Java WAR. Perhatikan bahwa AWS OpsWorks Stacks tidak mendukung tarball yang tidak terkompresi. Untuk menyebarkan beberapa halaman JSP, sertakan mereka dalam arsip yang sama.
3. Unggah arsip ke bucket Amazon S3 dan buat file menjadi publik. Salin URL file untuk digunakan nanti. Untuk informasi selengkapnya tentang cara membuat bucket dan mengunggah file, buka Memulai Dengan Amazon Simple Storage Service.
4. Tambahkan aplikasi ke tumpukan dan tentukan pengaturan berikut:
   - Nama – `SimpleJSP`
   - Jenis aplikasi - `Java`

- Jenis repositori - `Http Archive`
- URL Repositori — URL Amazon S3 dari file arsip Anda.

Gunakan nilai default untuk setelan yang tersisa, lalu klik Tambah Aplikasi untuk membuat aplikasi.

5. Menerapkan aplikasi ke instance Java App Server.

Anda sekarang dapat pergi ke URL aplikasi dan melihat aplikasi. Jika Anda belum menentukan domain, Anda dapat membuat URL dengan menggunakan alamat IP publik instans atau nama DNS publiknya. Untuk mendapatkan alamat IP publik instans atau nama DNS publik, buka konsol AWS OpsWorks Stacks dan klik nama instans pada halaman Instances untuk membuka halaman detailnya.

URL lainnya bergantung pada nama pendek aplikasi, yang merupakan nama huruf kecil yang dihasilkan AWS OpsWorks Stacks dari nama aplikasi yang Anda tentukan saat membuat aplikasi. Misalnya nama pendek SimpleJSP adalah simplejsp. Anda bisa mendapatkan nama pendek aplikasi dari halaman detailnya.

- Jika nama singkatnya`root`, Anda dapat menggunakan salah satu `http://`*`public_DNS`*`/`*`appname`*`.jsp` atau`http://`*`public_IP`*`/`*`appname`*`.jsp`.
- Jika tidak, Anda dapat menggunakan salah satu `http://`*`public_DNS`*`/`*`app_shortname`*`/`*`appname`*`.jsp` atau`http://`*`public_IP`*`/`*`app_shortname`*`/`*`appname`*`.jsp`.

Jika Anda telah menentukan domain untuk aplikasi, URL-nya adalah`http://`*`domain`*`/`*`appname`*`.jsp`.

URL untuk contoh akan menjadi sesuatu seperti`http://192.0.2.0/simplejsp/simplejsp.jsp`.

Jika Anda ingin menerapkan beberapa aplikasi ke instance yang sama, sebaiknya Anda tidak menggunakan `root` nama pendek. Ini dapat menyebabkan konflik URL yang mencegah aplikasi berfungsi dengan baik. Sebagai gantinya, tetapkan nama domain yang berbeda untuk setiap aplikasi.

Menyebarkan Aplikasi JSP dengan Back-End Database

Halaman JSP dapat menggunakan `DataSource` objek JDBC untuk terhubung ke database back end. Anda membuat dan menerapkan aplikasi semacam itu dengan menggunakan prosedur di bagian sebelumnya, dengan satu langkah tambahan untuk mengatur koneksi.

Halaman JSP berikut menunjukkan bagaimana menghubungkan ke `DataSource` objek.

```
<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 membuat dan menginisialisasi `DataSource` objek, mengikatnya ke nama logis, dan mendaftarkan nama dengan layanan penamaan Java Naming and Directory Interface (JNDI). Nama logis lengkapnya adalah`java:comp/env/`*`user-assigned-name`*. Anda harus menentukan bagian nama yang ditetapkan pengguna dengan menambahkan atribut JSON khusus ke konfigurasi tumpukan dan atribut penerapan untuk menentukan `['opsworks_java']['datasources']` atribut, seperti yang dijelaskan di bawah ini.

Untuk menyebarkan halaman JSP yang terhubung ke database MySQL

1. Buat tumpukan dengan lapisan Java App Server, tambahkan instance 24/7 ke setiap lapisan, dan mulai.
2. Tambahkan lapisan database ke tumpukan. Detailnya tergantung pada database mana yang Anda gunakan.

   Untuk menggunakan instance MySQL sebagai contoh, tambahkan layer MySQL ke tumpukan, tambahkan instance 24/7 ke layer, dan mulai.

   Untuk menggunakan instance Amazon RDS (MySQL) sebagai contoh:

   - Tentukan mesin database MySQL untuk contoh tersebut.
   - Tetapkan grup keamanan AWS- OpsWorks -DB-Master-Server (***`security_group_id`***) dan AWS OpsWorks - ***`security_group_id`*** -Java-App-Server () ke instance. AWS OpsWorks

Stacks membuat grup keamanan ini untuk Anda saat Anda membuat tumpukan pertama di wilayah tersebut.

- Buat database bernama`simplejspdb`.
- Pastikan bahwa nama pengguna dan kata sandi master tidak mengandung & atau karakter lain yang dapat menyebabkan kesalahan Tomcat.

  Khususnya selama startup Tomcat harus mengurai file konteks aplikasi web, yang merupakan file XMLyang menyertakan kata sandi utama dan nama pengguna. Jika salah satu string menyertakan & karakter, parser XMLmemperlakukannya sebagai entitas XHTML yang cacat dan melempar pengecualian parsing, yang mencegah Tomcat memulai. Untuk informasi selengkapnya tentang file konteks aplikasi web, lihatkucing jantan: :konteks.
- Tambahkan driver MySQL ke lapisan Java App Server.
- Daftarkan instance RDS dengan tumpukan Anda.

Untuk informasi selengkapnya tentang cara menggunakan instans Amazon RDS dengan AWS OpsWorks Stacks, lihat. Lapisan Layanan Amazon RDS

3. Salin kode contoh ke file bernama`simplejspdb.jsp`, letakkan file di folder bernama`simplejspdb`, dan buat `.zip` arsip folder. Namanya sewenang-wenang; Anda dapat menggunakan nama file atau folder apa pun yang Anda inginkan. Anda juga dapat menggunakan jenis arsip lain, termasuk gzip, bzip2, atau tarball. Untuk menyebarkan beberapa halaman JSP, sertakan mereka dalam arsip yang sama. Untuk informasi tentang cara menerapkan aplikasi dari jenis repositori lain, lihat. Sumber Aplikasi
4. Unggah arsip ke bucket Amazon S3 dan buat file menjadi publik. Salin URL file untuk digunakan nanti. Untuk informasi selengkapnya tentang cara membuat bucket dan mengunggah file, buka Memulai Dengan Amazon Simple Storage Service.
5. Tambahkan aplikasi ke tumpukan dan tentukan pengaturan berikut:

   - Nama – `SimpleJSPDB`
   - Jenis aplikasi - `Java`
   - Jenis sumber data — OpsWorks(untuk instance MySQL) atau RDS (untuk instans Amazon RDS).
   - Instance Database - Instance MySQL yang Anda buat sebelumnya, yang biasanya bernama db-master1 (mysql), atau instance Amazon RDS, yang akan diberi nama (mysql). ***`DB_instance_name`***
   - Nama basis data —`simplejspdb`.

- Jenis repositori - `Http Archive`
- URL Repositori — URL Amazon S3 dari file arsip Anda.

Gunakan nilai default untuk setelan yang tersisa, lalu klik Tambah Aplikasi untuk membuat aplikasi.

6. Tambahkan atribut JSON kustom berikut ke atribut konfigurasi tumpukan, di mana simplejspdb adalah nama pendek aplikasi.

```
{
  "opsworks_java": {
    "datasources": {
      "simplejspdb": "jdbc/mydb"
    }
  }
}
```

AWS OpsWorks Stacks menggunakan pemetaan ini untuk menghasilkan file konteks dengan informasi database yang diperlukan.

Untuk informasi selengkapnya tentang cara menambahkan atribut JSON kustom ke atribut konfigurasi tumpukan, lihatMenggunakan JSON Kustom.

7. Menerapkan aplikasi ke instance Java App Server.

Anda sekarang dapat menggunakan URL aplikasi untuk melihat aplikasi. Untuk deskripsi tentang cara membuat URL, lihatMenerapkan Aplikasi JSP.

URL untuk contoh akan menjadi sesuatu sepertihttp://192.0.2.0/simplejspdb/simplejspdb.jsp.

> **Note**
>
> `datasources`Atribut dapat berisi beberapa atribut. Setiap atribut diberi nama dengan nama pendek aplikasi dan disetel ke bagian yang ditetapkan pengguna yang sesuai dari nama logis. Jika Anda memiliki beberapa aplikasi, Anda dapat menggunakan nama logis terpisah, yang memerlukan JSON khusus seperti berikut ini.

```
{
  "opsworks_java": {
    "datasources": {
      "myjavaapp": "jdbc/myappdb",
      "simplejsp": "jdbc/myjspdb",
      ...
    }
  }
}
```

Lapisan AWS OpsWorks Tumpukan Server Aplikasi Node.js

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan Server Aplikasi Node.js adalah lapisan AWS OpsWorks Stacks yang menyediakan cetak biru untuk instance yang berfungsi sebagai server aplikasi Node.js. AWS OpsWorks Stacks juga menginstal Express, sehingga instance layer mendukung aplikasi standar dan Express.

Instalasi: Node.js diinstal di`/usr/local/bin/node`.

Halaman Add Layer menyediakan opsi konfigurasi berikut:

Versi Node.js

Untuk daftar versi yang saat ini didukung, lihatAWS OpsWorks Stacks sistem operasi.

Grup keamanan kustom

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Elastic Load Balancer

Anda dapat memasang penyeimbang beban Elastic Load Balancing ke instance layer.

> **Important**
>
> Jika aplikasi Node.js Anda menggunakan SSL, kami sarankan Anda menonaktifkan SSLv3 jika memungkinkan untuk mengatasi kerentanan yang dijelaskan dalam CVE-2015-8027. Untuk melakukannya, Anda harus mengatur versi Node.js ke`0.12.9`.

Menerapkan Aplikasi Node.js

Untuk panduan mendetail tentang cara menerapkan aplikasi Node.js sederhana untuk AWS OpsWorks Stacks dan menerapkannya ke tumpukan, lihat. Membuat Stack Node.js Pertama Anda Secara umum, aplikasi Node.js untuk AWS OpsWorks Stacks harus memenuhi ketentuan berikut:

- File utama harus diberi nama `server.js` dan berada di direktori root aplikasi yang digunakan.
- Aplikasi ekspres harus menyertakan `package.json` file di direktori root aplikasi.
- Secara default, aplikasi harus mendengarkan pada port 80 (HTTP) atau port 443 (HTTPS).

  Dimungkinkan untuk mendengarkan di port lain, tetapi grup keamanan bawaan lapisan Node.js App Server, AWS- OpsWorks -NodeJS-App-Server, memungkinkan lalu lintas pengguna masuk hanya ke port 80, 443, dan 22 (SSH). Untuk mengizinkan lalu lintas pengguna masuk ke port lain, buat grup keamanan dengan aturan masuk yang sesuai dan tetapkan ke lapisan Server Aplikasi Node.js. Jangan mengubah aturan masuk dengan mengedit grup keamanan bawaan. Setiap kali Anda membuat tumpukan, AWS OpsWorks Stacks menimpa grup keamanan bawaan dengan pengaturan standar, sehingga setiap perubahan yang Anda buat akan hilang.

> **Note**
>
> AWS OpsWorks Stacks menetapkan variabel lingkungan PORT ke 80 (default) atau 443 (jika Anda mengaktifkan SSL), sehingga Anda dapat menggunakan kode berikut untuk mendengarkan permintaan.
>
> ```
> app.listen(process.env.PORT);
> ```

Jika Anda mengonfigurasi aplikasi Node.js untuk mendukung SSL, Anda harus menentukan kunci dan sertifikat. AWS OpsWorks Stacks menempatkan data untuk setiap instance server aplikasi sebagai file terpisah dalam `/srv/www/app_shortname/shared/config` direktori, sebagai berikut.

- `ssl.crt`Sertifikat SSL
- `ssl.key`— kunci SSL.
- `ssl.ca`— sertifikat rantai, jika Anda telah menentukan satu.

Aplikasi Anda dapat memperoleh kunci SSL dan sertifikat dari file-file tersebut.

Lapisan AWS OpsWorks Tumpukan Server Aplikasi PHP

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan PHP App Server adalah lapisan AWS OpsWorks Stacks yang menyediakan cetak biru untuk instance yang berfungsi sebagai server aplikasi PHP. Lapisan PHP App Server didasarkan pada Apache2 dengan `mod_php` dan tidak memiliki opsi konfigurasi standar. Versi PHP dan Apache tergantung pada sistem operasi yang Anda tentukan untuk instance layer.

Sistem Operasi	Versi PHP	Versi 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

Instalasi: AWS OpsWorks Stacks menggunakan penginstal paket instans untuk menginstal Apache2 dan `mod_php` di lokasi defaultnya. Untuk informasi lebih lanjut tentang instalasi, lihat Apache.

Halaman Add Layer menyediakan opsi konfigurasi berikut:

Grup keamanan kustom

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Elastic Load Balancer

Anda dapat memasang penyeimbang beban Elastic Load Balancing ke instance layer.

Anda dapat memodifikasi beberapa pengaturan konfigurasi Apache dengan menggunakan JSON kustom atau file atribut kustom. Untuk informasi selengkapnya, lihat Mengesampingkan Atribut. Untuk daftar atribut Apache yang dapat diganti, lihat. Atribut apache2

Untuk contoh cara menerapkan Aplikasi PHP, termasuk cara menghubungkan aplikasi ke database backend, lihat. Memulai dengan Chef 11 Linux Stacks

> **Important**
>
> Jika aplikasi PHP Anda menggunakan SSL, kami sarankan Anda menonaktifkan SSLv3 jika mungkin untuk mengatasi kerentanan yang dijelaskan dalam CVE-2014-3566. Untuk melakukannya, Anda harus mengubah `SSLProtocol` pengaturan di `ssl.conf` file server Apache. Untuk informasi selengkapnya tentang cara mengubah setelan ini, lihatMenonaktifkan Server SSLv3 Apache.

Lapisan AWS OpsWorks Tumpukan Server Aplikasi Rails

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Layer Rails App Server adalah lapisan AWS OpsWorks Stacks yang menyediakan cetak biru untuk instance yang berfungsi sebagai server aplikasi Rails.

Instalasi: AWS OpsWorks Stacks menggunakan penginstal paket instance untuk menginstal paket server di lokasi defaultnya. Untuk informasi lebih lanjut tentang Apache/Passenger pemasangan, lihat Penumpang Phusion. Untuk informasi selengkapnya tentang pencatatan, lihat File Log. Untuk informasi lebih lanjut tentang Nginx/Unicorn instalasi, lihat Unicorn.

Halaman Add Layer menyediakan opsi konfigurasi berikut, yang semuanya opsional.

Versi Ruby

Versi Ruby yang akan digunakan oleh aplikasi Anda. Nilai defaultnya adalah 2.3.

Anda juga dapat menentukan versi Ruby pilihan Anda dengan mengganti atribut. `[:opsworks][:ruby_version]`

> **Note**
>
> AWS OpsWorks Stacks menginstal paket Ruby terpisah untuk digunakan oleh resep dan agen instance. Untuk informasi selengkapnya, lihat Versi Ruby.

Tumpukan Rel

Tumpukan Rails default adalah Apache2 dengan Phusion Passenger. Anda juga dapat menggunakan Nginx dengan Unicorn.

> **Note**
>
> Jika Anda menggunakan Nginx dan Unicorn, Anda harus menambahkan permata unicorn ke Gemfile aplikasi Anda, seperti pada contoh berikut:
>
> ```
> source 'https://rubygems.org'
> gem 'rails', '3.2.15'
> ...
> # Use unicorn as the app server
> gem 'unicorn'
> ...
> ```

Versi Penumpang

Jika Anda telah menentukan Apache2/Passenger, Anda harus menentukan versi Penumpang. Nilai defaultnya adalah 5.0.28.

Versi Rubygems

Versi Rubygems default adalah 2.5.1

Instal dan Kelola Bundler

Memungkinkan Anda memilih apakah akan menginstal dan mengelola Bundler. Nilai defaultnya adalah Ya.

Versi Bundler

Versi Bundler default adalah 1.12.5.

Grup keamanan kustom

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Elastic Load Balancer

Anda dapat memasang penyeimbang beban Elastic Load Balancing ke instance layer.

Anda dapat mengubah beberapa pengaturan konfigurasi dengan menggunakan JSON kustom atau file atribut kustom. Untuk informasi selengkapnya, lihat Mengesampingkan Atribut. Untuk daftar atribut Apache, Nginx, Phusion Passenger, dan Unicorn yang dapat diganti, lihat. Atribut Buku Masak Bawaan

> **Important**
>
> Jika aplikasi Ruby on Rails Anda menggunakan SSL, kami sarankan Anda menonaktifkan SSLv3 jika memungkinkan untuk mengatasi kerentanan yang dijelaskan dalam CVE-2014-3566. Untuk informasi selengkapnya, lihat Menonaktifkan SSLv3 untuk Rails Server.

Topik

- Menonaktifkan SSLv3 untuk Rails Server
- Menghubungkan ke Database
- Menyebarkan Ruby di Aplikasi Rails

Menonaktifkan SSLv3 untuk Rails Server

SSLv3 Untuk menonaktifkan server Rails, perbarui pengaturan Ruby Version layer ke 2.1 atau lebih tinggi, yang menginstal Ruby 2.1.4 atau lebih tinggi sebagai versi yang digunakan aplikasi.

- Perbarui pengaturan Ruby Version layer ke 2.1 atau lebih tinggi.
- Perbarui file konfigurasi untuk tumpukan Rails Anda, sebagai berikut.

Apache dengan Penumpang Phusion

Perbarui `SSLProtocol` pengaturan dalam `ssl.conf` file server Apache, seperti yang dijelaskan dalamMenonaktifkan Server SSLv3 Apache.

Nginx dengan Unicorn

Tambahkan `ssl_protocols` direktif eksplisit ke file server Nginx. `nginx.conf` Untuk menonaktifkan SSLv3, ganti file template buku masak `nginx.conf.erb` nginx bawaan, yang digunakan resep Pengaturan lapisan Rails App Server untuk `nginx.conf` membuat, dan tambahkan arahan berikut:

```
ssl_protocols TLSv1.2;
```

Untuk informasi selengkapnya tentang cara mengonfigurasi`nginx.conf`, lihat Mengonfigurasi server HTTPS. Untuk informasi selengkapnya tentang cara mengganti template bawaan, lihatMenggunakan Template Kustom.

Menghubungkan ke Database

Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks membuat `database.yml` file baru menggunakan informasi dari atribut aplikasi. `deploy` Jika Anda melampirkan instance MySQL atau Amazon RDS ke AWS OpsWorks aplikasi, Stacks menambahkan informasi koneksi `deploy` ke atribut, `database.yml` sehingga secara otomatis berisi data koneksi yang benar.

Jika aplikasi tidak memiliki database terlampir, secara default, AWS OpsWorks Stacks tidak menambahkan informasi koneksi apa pun ke `deploy` atribut dan tidak membuat`database.yml`. Jika ingin menggunakan database yang berbeda, Anda dapat menggunakan JSON khusus untuk menambahkan atribut database ke `deploy` atribut aplikasi dengan informasi koneksi. Semua atribut ada di bawah`["deploy"]["`*`appshortname`*`"]["database"]`, di *`appshortname`* mana

nama pendek aplikasi, yang dihasilkan AWS OpsWorks Stacks dari nama aplikasi. Nilai yang Anda tentukan di JSON kustom mengganti pengaturan default apa pun. Untuk informasi selengkapnya, lihat Menambahkan Aplikasi.

AWS OpsWorks Tumpukan menggabungkan nilai `[:...][:database]`atribut berikut ke dalam. `database.yml` Atribut yang diperlukan tergantung pada database tertentu, tetapi Anda harus memiliki `host` atribut atau AWS OpsWorks Stacks tidak akan membuat`database.yml`.

- `[:adapter] (String)`— Adaptor basis data, sepertimysql.
- `[:database]`(String) — Nama database.
- `[:encoding]`(String) — Pengkodean, yang biasanya diatur keutf8.
- `[:host]`(String) — URL host, sepertirailsexample.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com.
- `[:reconnect]`(Boolean) - Apakah aplikasi harus terhubung kembali jika koneksi tidak ada lagi.
- `[:password]`(String) — Kata sandi database.
- `[:port]`(Jumlah). — Nomor port database. Gunakan atribut ini untuk mengganti nomor port default, yang diatur oleh disetel oleh adaptor.
- `[:username]`(String) - Nama pengguna database.

Contoh berikut menunjukkan JSON kustom untuk aplikasi yang nama pendeknya adalah myapp.

```
{
  "deploy" : {
    "myapp" : {
      "database" : {
        "adapter" : "adapter",
        "database" : "databasename",
        "host" : "host",
        "password" : "password",
        "port" : portnumber
        "reconnect" : true/false,
        "username" : "username"
      }
    }
  }
}
```

Untuk informasi tentang cara menentukan JSON kustom, lihatMenggunakan JSON Kustom. Untuk melihat template yang digunakan untuk create `database.yml` (`database.yml.erb`), buka repositori buku masak bawaan.

Menyebarkan Ruby di Aplikasi Rails

Anda dapat menerapkan aplikasi Ruby on Rails dari salah satu repositori yang didukung. Berikut ini menunjukkan cara menerapkan contoh aplikasi Ruby on Rails ke server yang menjalankan tumpukan Rails. Apache/Passenger Kode contoh disimpan dalam GitHub repositori publik, tetapi prosedur dasarnya sama untuk repositori lain yang didukung. Untuk informasi selengkapnya tentang cara membuat dan menerapkan aplikasi, lihatAplikasi. Untuk melihat kode contoh, yang menyertakan komentar ekstensif, buka https://github.com/awslabs/opsworks-demo-rails-photo-share-app.

Untuk menerapkan aplikasi Ruby on Rails dari repositori GitHub

1. Buat tumpukan dengan layer Rails App Server dengan Apache/Passenger sebagai tumpukan Rails, tambahkan instance 24/7 ke layer, dan mulai.
2. Setelah instance online, tambahkan aplikasi ke tumpukan dan tentukan pengaturan berikut:

   - Nama — Nama apa pun yang Anda inginkan; contohnya menggunakan`PhotoPoll`.

     AWS OpsWorks Stacks menggunakan nama ini untuk tujuan tampilan, dan menghasilkan nama pendek untuk penggunaan internal dan untuk mengidentifikasi aplikasi dalam konfigurasi tumpukan dan atribut penerapan. Misalnya, nama PhotoPoll pendeknya adalah photopoll.
   - Jenis aplikasi — Ruby on Rails.
   - Lingkungan Rails — Lingkungan yang tersedia ditentukan oleh aplikasi.

     Aplikasi contoh memiliki tiga:**`development`**,**`test`**, dan**`production`**. Untuk contoh ini, atur lingkungan ke**`development`**. Lihat kode contoh untuk deskripsi setiap lingkungan.
   - Jenis repositori — Salah satu jenis repositori yang didukung. Tentukan `Git` untuk contoh ini
   - URL Repositori — Repositori tempat kode harus digunakan.

     Untuk contoh ini, atur URL ke**`git://github.com/awslabs/opsworks-demo-rails-photo-share-app`**.

   Gunakan nilai default untuk setelan yang tersisa, lalu klik Tambah Aplikasi untuk membuat aplikasi.

3. Menerapkan aplikasi ke instance Rails App Server.
4. Saat penerapan selesai, buka halaman Instans dan klik alamat IP publik instans Rails App Server. Anda akan melihat yang berikut ini:



Lapisan AWS OpsWorks Tumpukan Server Web Statis

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Layer Static Web Server adalah lapisan AWS OpsWorks Stacks yang menyediakan template untuk instance untuk melayani halaman HTML statis, yang dapat mencakup scripting sisi klien. Lapisan ini didasarkan pada Nginx.

Instalasi: Nginx dipasang di. `/usr/sbin/nginx`

Halaman Add Layer menyediakan opsi konfigurasi berikut:

Grup keamanan kustom

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Elastic Load Balancer

Anda dapat memasang penyeimbang beban Elastic Load Balancing ke instance layer.

Anda dapat memodifikasi beberapa pengaturan konfigurasi Nginx dengan menggunakan JSON kustom atau file atribut kustom. Untuk informasi selengkapnya, lihat Mengesampingkan Atribut. Untuk daftar atribut Apache yang dapat diganti, lihat. nginx Atribut

Important

Jika aplikasi web Anda menggunakan SSL, sebaiknya Anda menonaktifkan SSLv3 jika memungkinkan untuk mengatasi kerentanan yang dijelaskan dalam CVE-2014-3566.
Untuk menonaktifkan SSLv3, Anda harus memodifikasi file server Nginx. `nginx.conf` Untuk melakukannya, ganti file template buku masak `nginx.conf.erb` nginx bawaan, yang digunakan resep Pengaturan lapisan Rails App Server untuk `nginx.conf` membuat, dan tambahkan arahan berikut:

```
ssl_protocols TLSv1.2;
```

Untuk informasi selengkapnya tentang cara mengonfigurasi`nginx.conf`, lihat Mengonfigurasi server HTTPS. Untuk informasi selengkapnya tentang cara mengganti template bawaan, lihatMenggunakan Template Kustom.

Referensi Lapisan Cluster ECS

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan ECS Cluster mewakili klaster Amazon Elastic Container Service (Amazon ECS) dan menyederhanakan pengelolaan klaster.

Nama pendek: ecs-cluster

Kompatibilitas: Lapisan layanan Amazon ECS hanya kompatibel dengan lapisan khusus

Port terbuka: ECS Cluster memungkinkan akses publik ke port 22 (SSH)

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Tidak

Grup keamanan default: AWS-OpsWorks-ECS-Cluster

Konfigurasi: Untuk mengkonfigurasi lapisan ECS Cluster, Anda harus menentukan yang berikut:

- Apakah akan menetapkan alamat IP publik atau alamat IP Elastis ke instance container
- Profil instance untuk instance kontainer

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- opsworks_ecs: :pengaturan

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- mysql: :klien
- agent_version
- opsworks_ecs: :konfigurasi

Menyebarkan resep:

- menyebarkan: :default
- opsworks_ecs: :menyebarkan

Undeploy resep:

- opsworks_ecs: :undeploy

Resep shutdown:

- opsworks_shutdown: :default
- opsworks_ecs: :shutdown

Instalasi:

- AWS OpsWorks Stacks menggunakan penginstal paket instance untuk menginstal Docker ke lokasi defaultnya
- Log Chef untuk acara Pengaturan mencatat apakah agen Amazon ECS berhasil diinstal. Jika tidak, log yang disediakan oleh AWS OpsWorks Stacks tidak menyertakan informasi log kesalahan Amazon ECS. Untuk informasi selengkapnya tentang cara menangani error Amazon ECS, lihat Pemecahan Masalah Amazon ECS.

Referensi Lapisan Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika lapisan standar tidak sesuai dengan kebutuhan Anda, Anda dapat membuat lapisan khusus. Tumpukan dapat memiliki beberapa lapisan khusus. Secara default, lapisan kustom menjalankan serangkaian resep standar terbatas yang mendukung fungsionalitas dasar. Anda kemudian mengimplementasikan fungsionalitas utama layer dengan menerapkan satu set resep Chef kustom untuk setiap peristiwa siklus hidup yang sesuai untuk mengatur dan mengkonfigurasi perangkat lunak lapisan, dan sebagainya. Resep khusus berjalan setelah resep AWS OpsWorks Stacks standar untuk setiap acara.

Nama pendek: Ditentukan pengguna; setiap lapisan kustom dalam tumpukan harus memiliki nama pendek yang berbeda

Port terbuka: Secara default, lapisan server kustom membuka akses publik ke port 22 (SSH), 80 (HTTP), 443 (HTTPS), dan semua port dari lapisan server aplikasi Rails dan PHP stack

Autoassign Alamat IP Elastis: Nonaktif secara default

Volume EBS default: Tidak

Grup Keamanan Default: AWS-OpsWorks-Custom-Server

Kompatibilitas: Lapisan khusus kompatibel dengan lapisan berikut: custom, db-master, lb, memcached, monitoring-master, nodejs-app, php-app, rails-app, dan web

Konfigurasi: Untuk mengkonfigurasi lapisan khusus, Anda harus menentukan yang berikut:

- Nama layer
- Nama pendek layer, yang mengidentifikasi lapisan dalam resep Chef dan harus menggunakan hanya a-z dan angka

Untuk tumpukan Linux, lapisan kustom menggunakan resep berikut.

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys

- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version

Menyebarkan resep:

- menyebarkan: :default

Resep shutdown:

- opsworks_shutdown: :default

Referensi Lapisan Lainnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Tumpukan juga mendukung lapisan berikut.

Topik

- Referensi Lapisan Ganglia

- Referensi Lapisan Memcached

Referensi Lapisan Ganglia

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Lapisan Ganglia mendukung Ganglia, sistem pemantauan terdistribusi yang mengelola penyimpanan dan visualisasi metrik instance. Ini dirancang untuk bekerja dengan topologi instance hierarkis, yang membuatnya sangat berguna untuk kelompok instance. Ganglia memiliki dua komponen dasar:

- Klien overhead rendah, yang diinstal pada setiap instance di tumpukan dan mengirimkan metrik ke master.
- Master, yang mengumpulkan metrik dari klien dan menyimpannya di volume Amazon EBS. Ini juga menampilkan metrik pada halaman web.

AWS OpsWorks Stacks memiliki agen pemantauan Ganglia pada setiap instance yang dikelola. Saat Anda menambahkan layer Ganglia ke tumpukan Anda dan memulainya, agen Ganglia pada setiap instance melaporkan metrik ke instance Ganglia. Untuk menggunakan Ganglia, tambahkan layer Ganglia dengan satu instance ke tumpukan. Anda mengakses data dengan masuk ke backend Ganglia di alamat IP master. Anda dapat memberikan definisi metrik tambahan dengan menulis resep Chef.

Nama pendek: monitoring-master

Kompatibilitas: Lapisan Ganglia kompatibel dengan lapisan berikut: custom, db-master, memcached, php-app, rails-app.

Port terbuka: Load-Balancer memungkinkan akses publik ke port 22 (SSH), 80 (HTTP), dan 443 (HTTPS).

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Ya, di `/vol/ganglia`

Grup keamanan default: AWS-OpsWorks-Monitoring-Master -Server

Konfigurasi: Untuk mengkonfigurasi lapisan Ganglia, Anda harus menentukan yang berikut:

- URI yang menyediakan akses ke grafik pemantauan. Nilai defaultnya adalah http://*DNSName*/ganglia, di mana nama *DNSName* DNS instance Ganglia.
- Nama pengguna dan kata sandi yang mengontrol akses ke statistik pemantauan.

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- opsworks_ganglia: :server

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version
- opsworks_ganglia: :configurre-server

Menyebarkan resep:

- menyebarkan: :default

- opsworks_ganglia: :configurre-server
- opsworks_ganglia: :menyebarkan

Resep shutdown:

- opsworks_shutdown: :default
- apache2: :berhenti

Instalasi:

- Klien Ganglia dipasang di bawah:`/etc/ganglia`.
- Ujung depan web Ganglia dipasang di bawah:`/usr/share/ganglia-webfrontend`.
- Loggtailer Ganglia dipasang di bawah:. `/usr/share/ganglia-logtailer`

Referensi Lapisan Memcached

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk tumpukan berbasis Linux.

Memcached adalah sistem caching memori terdistribusi untuk data arbitrer. Ini mempercepat situs web dengan menyimpan string dan objek sebagai kunci dan nilai dalam RAM untuk mengurangi berapa kali sumber data eksternal harus dibaca.

Untuk menggunakan Memcached dalam tumpukan, buat layer Memcached dan menambahkan satu atau beberapa instance, yang berfungsi sebagai server Memcached. Instans secara otomatis

menginstal Memcached dan instance stack lainnya dapat mengakses dan menggunakan server Memcached. Jika Anda menggunakan layer Rails App Server, AWS OpsWorks Stacks secara otomatis menempatkan file `memcached.yml` konfigurasi di direktori konfigurasi setiap instance di layer. Anda dapat memperoleh server dan nomor port Memcached dari file ini.

Nama pendek: memcached

Kompatibilitas: Lapisan Memcached kompatibel dengan lapisan berikut: custom, db-master, lb, monitoring-master, nodejs-app, php-app, rails-app, dan web.

Port terbuka: Lapisan Memcached memungkinkan akses publik ke port 22 (SSH) dan semua port dari server web stack, server khusus, dan server aplikasi Rails, PHP, dan Node.js.

Autoassign alamat IP Elastis: Nonaktif secara default

Volume EBS default: Tidak

Grup keamanan default: AWS-OpsWorks-Memcached-Server

Untuk mengkonfigurasi layer Memcached, Anda harus menentukan ukuran cache, dalam MB.

Resep pengaturan:

- opsworks_initial_setup
- ssh_host_keys
- ssh_users
- mysql: :klien
- dependensi
- ebs
- opsworks_ganglia: :klien
- memcached

Konfigurasikan resep:

- opsworks_ganglia: :configurre-klien
- ssh_users
- agent_version

Menyebarkan resep:

- menyebarkan: :default

Resep shutdown:

- opsworks_shutdown: :default
- memcached: :berhenti

Instalasi:

- AWS OpsWorks Stacks menggunakan penginstal paket instance untuk menginstal Memcached dan file lognya di lokasi defaultnya.

Lapisan lainnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk Chef 11 dan tumpukan berbasis Linux sebelumnya.

AWS OpsWorks Stacks juga mendukung lapisan Ganglia dan Memcached.

Topik

- Lapisan Ganglia
- Memcached

Lapisan Ganglia

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk Chef 11 dan tumpukan berbasis Linux sebelumnya.

AWS OpsWorks Stacks mengirimkan semua metrik instans dan volume Anda ke Amazon CloudWatch, sehingga memudahkan untuk melihat grafik dan menyetel alarm untuk membantu Anda memecahkan masalah dan mengambil tindakan otomatis berdasarkan status sumber daya Anda. Anda juga dapat menggunakan lapisan Ganglia AWS OpsWorks Stacks untuk opsi pemantauan aplikasi tambahan seperti menyimpan metrik pilihan Anda.

Lapisan Ganglia adalah cetak biru untuk instance yang memantau tumpukan Anda dengan menggunakan pemantauan terdistribusi Ganglia. Sebuah tumpukan biasanya hanya memiliki satu contoh Ganglia. Lapisan Ganglia mencakup pengaturan konfigurasi opsional berikut:

URL Ganglia

Jalur URL statistik. URL lengkapnya adalah http://*DNSName*/*URLPath*, di *DNSName* mana nama DNS instance terkait. *URLPath*Nilai default adalah “/ganglia” yang sesuai dengan sesuatu seperti: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.

Nama pengguna Ganglia

Nama pengguna untuk halaman web statistik. Anda harus memberikan nama pengguna saat melihat halaman. Nilai defaultnya adalah “opsworks”.

Kata sandi Ganglia

Kata sandi yang mengontrol akses ke halaman web statistik. Anda harus memberikan kata sandi saat melihat halaman. Nilai default adalah string yang dihasilkan secara acak.

> **Note**
>
> Rekam kata sandi untuk digunakan nanti; AWS OpsWorks Tumpukan tidak memungkinkan Anda untuk melihat kata sandi setelah Anda membuat lapisan. Namun, Anda dapat memperbarui kata sandi dengan membuka halaman Edit layer dan mengklik Perbarui kata sandi.

Grup keamanan khusus

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Elastic Load Balancer

Anda dapat memasang penyeimbang beban Elastic Load Balancing ke instance layer.

> **Important**
>
> Jika tumpukan Anda menyertakan lapisan Ganglia, sebaiknya Anda menonaktifkan SSLv3 jika memungkinkan lapisan tersebut untuk mengatasi kerentanan yang dijelaskan dalam CVE-2014-3566. Untuk melakukannya, Anda harus mengganti `ssl.conf.erb` template server Apache untuk memodifikasi pengaturan. `SSLProtocol` Lihat perinciannya di Menonaktifkan Server SSLv3 Apache.

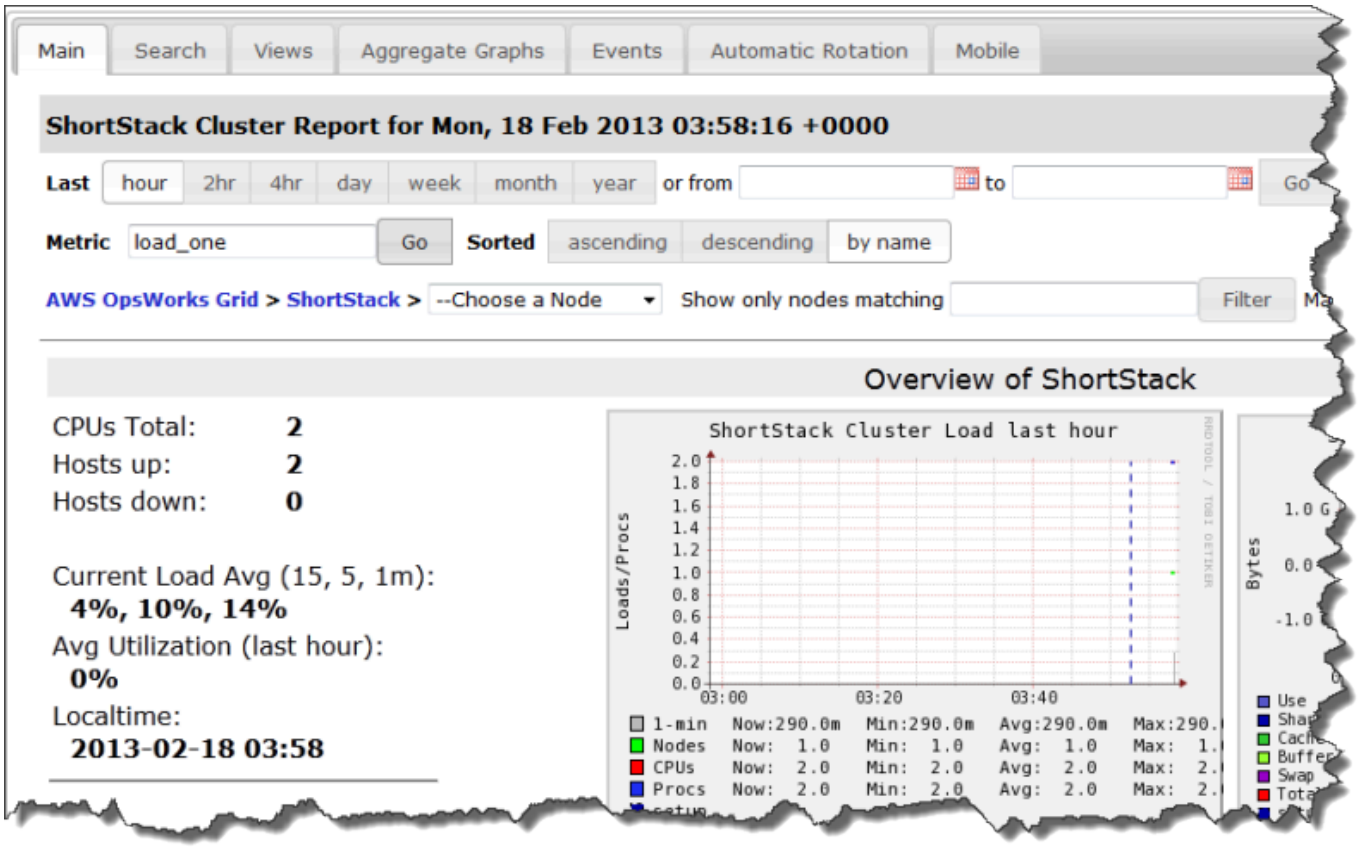
Lihat Statistik Ganglia

AWS OpsWorks Resep tumpukan menginstal klien Ganglia overhead rendah pada setiap instance. Jika tumpukan Anda menyertakan lapisan Ganglia, klien Ganglia secara otomatis mulai melapor ke Ganglia segera setelah instance muncul. Ganglia menggunakan data klien untuk menghitung berbagai statistik dan menampilkan hasilnya secara grafis pada halaman web statistiknya.

Untuk melihat statistik Ganglia

1. Pada halaman Layers, klik Ganglia untuk membuka halaman detail layer.
2. Di panel navigasi, klik Instans. Di bawah Ganglia, klik nama instance.

3. Salin nama DNS Publik instans.
4. Gunakan nama DNS untuk membuat URL statistik, yang akan terlihat seperti: http://ec2-54-245-151-7.us-west-2.compute.amazonaws.com/ganglia.
5. Tempel URL lengkap ke browser Anda, arahkan ke halaman, dan masukkan nama pengguna dan kata sandi Ganglia untuk menampilkan halaman. Berikut contohnya.



Memcached

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Lapisan ini hanya tersedia untuk Chef 11 dan tumpukan berbasis Linux sebelumnya.

Lapisan Memcached adalah lapisan AWS OpsWorks Stacks yang menyediakan cetak biru untuk instance yang berfungsi sebagai server Memcached — sistem caching memori terdistribusi untuk data arbitrer. Lapisan Memcached mencakup pengaturan konfigurasi berikut.

Memori yang dialokasikan (MB)

(Opsional) Jumlah memori cache (dalam MB) untuk setiap instance layer. Defaultnya adalah 512 MB.

Grup keamanan khusus

Pengaturan ini muncul jika Anda memilih untuk tidak secara otomatis mengaitkan grup keamanan AWS OpsWorks Stacks bawaan dengan lapisan Anda. Anda harus menentukan grup keamanan mana yang akan dikaitkan dengan layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Komponen Buku Masak

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Buku masak biasanya mencakup komponen dasar berikut:

- File atribut berisi satu set atribut yang mewakili nilai yang akan digunakan oleh resep dan template.
- File template adalah template yang digunakan resep untuk membuat file lain, seperti file konfigurasi.

File template biasanya memungkinkan Anda memodifikasi file konfigurasi dengan mengesampingkan atribut—yang dapat dilakukan tanpa menyentuh buku masak—alih-alih menulis ulang file konfigurasi. Praktek standar adalah bahwa setiap kali Anda berharap untuk mengubah file konfigurasi pada sebuah instance bahkan sedikit, Anda harus menggunakan file template.

- File resep adalah aplikasi Ruby yang mendefinisikan semua yang diperlukan untuk mengkonfigurasi sistem, termasuk membuat dan mengonfigurasi folder, menginstal dan mengonfigurasi paket, memulai layanan, dan sebagainya.

Buku masak tidak harus memiliki ketiga komponen. Pendekatan yang lebih sederhana untuk kustomisasi hanya memerlukan atribut atau file template. Selain itu, buku masak secara opsional dapat menyertakan jenis file lain, seperti definisi atau spesifikasi.

Bagian ini menjelaskan tiga komponen buku masak standar. Untuk informasi selengkapnya, terutama tentang cara menerapkan resep, lihat Opscode.

Topik

- Atribut
- Template
- Resep

Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Resep dan template bergantung pada berbagai nilai, seperti pengaturan konfigurasi. Daripada hardcode nilai-nilai tersebut secara langsung dalam resep atau template, Anda dapat membuat file atribut dengan atribut yang mewakili setiap nilai. Anda kemudian menggunakan atribut dalam resep atau templat Anda alih-alih nilai eksplisit. Keuntungan menggunakan atribut adalah Anda dapat

mengganti nilainya tanpa menyentuh buku masak. Untuk alasan ini, Anda harus selalu menggunakan atribut untuk menentukan jenis nilai berikut:

- Nilai yang mungkin bervariasi dari tumpukan ke tumpukan atau dengan waktu, seperti nama pengguna.

  Jika Anda hardcode nilai-nilai tersebut, Anda harus mengubah resep atau template setiap kali Anda perlu mengubah nilai. Dengan menggunakan atribut untuk menentukan nilai-nilai ini, Anda dapat menggunakan buku masak yang sama untuk setiap tumpukan dan hanya mengganti atribut yang sesuai.

- Nilai sensitif, seperti kata sandi atau kunci rahasia.

  Menempatkan nilai sensitif eksplisit dalam buku masak Anda dapat meningkatkan risiko paparan. Sebagai gantinya, tentukan atribut dengan nilai dummy dan timpa untuk menetapkan nilai sebenarnya. Cara terbaik untuk mengganti atribut tersebut adalah dengan JSON khusus. Untuk informasi selengkapnya, lihat Menggunakan JSON Kustom.

Untuk informasi selengkapnya tentang atribut dan cara menggantinya, lihatMengesampingkan Atribut.

Contoh berikut adalah bagian dari file atribut contoh.

```
...
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 mendefinisikan atribut dengan menggunakan sintaks berikut:

```
node.type["attribute"]["subattribute"]["..."]=value
```

Anda juga dapat menggunakan titik dua (:), sebagai berikut:

```
node.type[:attribute][:subattribute][:...]=value
```

Definisi atribut memiliki komponen berikut:

`node.`

`node.`Awalan adalah opsional dan biasanya dihilangkan, seperti yang ditunjukkan pada contoh.

`type`

Tipe mengatur apakah atribut dapat diganti. AWS OpsWorks Atribut tumpukan biasanya menggunakan salah satu dari jenis berikut:

- `default`adalah tipe yang paling umum digunakan, karena memungkinkan atribut untuk diganti.
- `normal`mendefinisikan atribut yang menggantikan salah satu nilai atribut AWS OpsWorks Stacks standar.

> **Note**
>
> Chef mendukung jenis tambahan, yang tidak diperlukan untuk AWS OpsWorks Stacks tetapi mungkin berguna untuk proyek Anda. Untuk informasi selengkapnya, lihat Tentang Atribut.

`attribute name`

Nama atribut menggunakan sintaks simpul Chef standar,`[:attribute][:subattribute][...]`. Anda dapat menggunakan nama apa pun yang Anda suka untuk atribut Anda. Namun, seperti yang dibahas dalamMengesampingkan Atribut, atribut buku masak khusus digabungkan ke dalam objek node instance, bersama dengan atribut dari konfigurasi tumpukan dan atribut penerapan, dan alat Ohai Chef. Nama konfigurasi yang umum digunakan seperti port atau pengguna mungkin muncul di berbagai buku masak.

Untuk menghindari tabrakan nama, konvensi adalah membuat nama atribut yang memenuhi syarat dengan setidaknya dua elemen, seperti yang ditunjukkan pada contoh. Elemen pertama harus unik dan biasanya didasarkan pada nama produk seperti Apache. Ini diikuti oleh satu atau lebih subattribut

yang mengidentifikasi nilai tertentu, seperti atau. `[:user]` `[:port]` Anda dapat menggunakan subattribut sebanyak yang sesuai untuk proyek Anda.

`value`

Atribut dapat diatur ke jenis nilai berikut:

- Sebuah string, sepertidefault`[:apache][:keepalive] = 'Off'`.
- Angka (tanpa tanda kutip) sepertidefault`[:apache][:timeout] = 120`.
- Nilai Boolean, yang bisa berupa `true` atau `false` (tidak ada tanda kutip).
- Daftar nilai, seperti `default[:apache][:listen_ports] = [ '80','443' ]`

File atribut adalah aplikasi Ruby, sehingga Anda juga dapat menggunakan sintaks node dan operator logis untuk menetapkan nilai berdasarkan atribut lainnya. Untuk informasi selengkapnya tentang cara mendefinisikan atribut, lihat Tentang Atribut s. Untuk contoh file atribut yang berfungsi, lihat buku masak bawaan AWS OpsWorks Stacks di opsworks-cookbooks. https://github.com/aws/

Template

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda mengkonfigurasi banyak paket dengan membuat file konfigurasi dan menempatkannya di direktori yang sesuai. Anda dapat menyertakan file konfigurasi di buku masak Anda dan menyalinnya ke direktori yang sesuai, tetapi pendekatan yang lebih fleksibel adalah meminta resep Anda membuat file konfigurasi dari templat. Salah satu keuntungan dari template adalah Anda dapat menggunakan atribut untuk menentukan nilai template. Ini memungkinkan Anda, misalnya, untuk memodifikasi file konfigurasi tanpa menyentuh buku masak dengan menggunakan JSON khusus untuk mengganti nilai atribut yang sesuai.

Template pada dasarnya memiliki konten dan struktur yang sama dengan file terkait. Berikut adalah contoh file,`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>
...
```

Contoh berikut adalah `httpd.conf` file yang dihasilkan untuk 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>
...
```

Sebagian besar teks template hanya disalin dari template ke `httpd.conf` file. Namun, `<%= ... %>` konten ditangani sebagai berikut:

- Chef menggantikan `<%= node[:attribute][:sub_attribute][:...]%>` dengan nilai atribut.

  Misalnya, `StartServers <%= node[:apache][:prefork][:startservers] %>` menjadi `StartServers 16` di`httpd.conf`.
- Anda dapat menggunakan `<%if-%>, <%else-%>, and <%end-%>` untuk memilih nilai secara kondisional.

  Contoh menetapkan jalur file yang berbeda untuk `accept.lock` tergantung pada platform.

> **Note**
>
> Anda tidak terbatas pada atribut dalam file atribut buku masak Anda. Anda dapat menggunakan atribut apapun dalam objek node instance ini. Misalnya, dihasilkan oleh alat Chef yang disebut Ohai dan juga dimasukkan ke dalam objek node. Untuk informasi lebih lanjut tentang atribut, lihatMengesampingkan Atribut.

Untuk informasi selengkapnya tentang template, termasuk cara menggabungkan kode Ruby, lihat Tentang Template.

Resep

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Resep adalah aplikasi Ruby yang mendefinisikan konfigurasi sistem. Mereka menginstal paket, membuat file konfigurasi dari template, menjalankan perintah shell, membuat file dan direktori, dan sebagainya. Anda biasanya memiliki AWS OpsWorks Stacks mengeksekusi resep secara otomatis

ketika peristiwa siklus hidup terjadi pada instance tetapi Anda juga dapat menjalankannya secara eksplisit kapan saja dengan menggunakan perintah stack Execute Recipes. Untuk informasi lebih lanjut, lihat Tentang Resep.

Resep biasanya sebagian besar terdiri dari serangkaian sumber daya, yang masing-masing mewakili keadaan yang diinginkan dari suatu aspek sistem. Setiap sumber daya mencakup seperangkat atribut yang menentukan keadaan yang diinginkan dan menentukan tindakan apa yang harus diambil. Chef mengaitkan setiap sumber daya dengan penyedia yang sesuai yang melakukan tindakan. Untuk informasi selengkapnya, lihat Referensi Sumber Daya dan Penyedia.

`package`Sumber daya membantu Anda mengelola paket perangkat lunak pada instance Linux. Contoh berikut menginstal paket 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 menggunakan penyedia paket yang sesuai untuk platform. Atribut sumber daya seringkali hanya diberi nilai, tetapi Anda dapat menggunakan operasi logis Ruby untuk melakukan tugas bersyarat. Contoh menggunakan `case` operator, yang digunakan `node[:platform]` untuk mengidentifikasi sistem operasi instance dan menetapkan `package_name` atribut yang sesuai. Anda dapat menyisipkan atribut ke dalam resep dengan menggunakan sintaks simpul Chef standar dan Chef menggantinya dengan nilai terkait. Anda dapat menggunakan atribut apa pun di objek node, bukan hanya atribut buku masak Anda.

Setelah menentukan nama paket yang sesuai, segmen kode diakhiri dengan `install` tindakan, yang menginstal paket. Tindakan lain untuk sumber daya ini termasuk `upgrade` dan`remove`. Untuk informasi lebih lanjut, lihat paket.

Seringkali berguna untuk memecah tugas instalasi dan konfigurasi yang kompleks menjadi satu atau lebih subtugas, masing-masing diimplementasikan sebagai resep terpisah, dan meminta resep

utama Anda menjalankannya pada waktu yang tepat. Contoh berikut menunjukkan baris kode yang mengikuti contoh sebelumnya:

```
include_recipe 'apache2::service'
```

Untuk memiliki resep mengeksekusi resep anak, gunakan `include_recipe` kata kunci, diikuti dengan nama resep. Resep diidentifikasi dengan menggunakan *`CookbookName`*`::`*`RecipeName`* sintaks Chef standar, di mana *`RecipeName`* menghilangkan ekstensi. `.rb`

> **Note**
>
> Sebuah `include_recipe` pernyataan secara efektif mengeksekusi resep pada saat itu dalam resep utama. Namun, apa yang sebenarnya terjadi adalah bahwa Chef mengganti setiap `include_recipe` pernyataan dengan kode resep yang ditentukan sebelum mengeksekusi resep utama.

`directory`Sumber daya mewakili direktori, seperti direktori yang berisi file paket. `default.rb`Sumber daya berikut membuat direktori log Linux.

```
directory node[:apache][:log_dir] do
    mode 0755
    action :create
end
```

Direktori log didefinisikan dalam salah satu file atribut buku masak. Sumber daya menentukan mode direktori sebagai 0755, dan menggunakan `create` tindakan untuk membuat direktori. Untuk informasi selengkapnya, lihat direktori. Anda juga dapat menggunakan sumber daya ini dengan instance Windows.

Sumber `execute` daya mewakili perintah, seperti perintah shell atau skrip. Contoh berikut menghasilkan file module.load.

```
execute 'generate-module-list' do
  if node[:kernel][:machine] == 'x86_64'
    libdir = 'lib64'
```

```
  else
    libdir = 'lib'
  end
  command "/usr/local/bin/apache2_module_conf_generate.pl /usr/#{libdir}/httpd/
modules /etc/httpd/mods-available"
  action :run
end
```

Sumber daya pertama-tama menentukan jenis CPU. `[:kernel][:machine]`adalah salah satu atribut otomatis yang dihasilkan Chef untuk mewakili berbagai properti sistem, tipe CPU dalam kasus ini. Kemudian menentukan perintah, skrip Perl dan menggunakan `run` tindakan untuk menjalankan skrip, yang menghasilkan file.load modul. Untuk informasi selengkapnya, lihat mengeksekusi.

`template`Sumber daya mewakili file — biasanya file konfigurasi — yang akan dihasilkan dari salah satu file template buku masak. Contoh berikut membuat file `httpd.conf` konfigurasi dari `apache2.conf.erb` template yang dibahas diTemplate.

```
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
```

Sumber daya menentukan nama dan lokasi file yang dihasilkan berdasarkan sistem operasi instance. Kemudian menentukan `apache2.conf.erb` sebagai template yang akan digunakan untuk menghasilkan file dan menetapkan pemilik, grup, dan mode file. Ini menjalankan `notify` tindakan untuk memberi tahu `service` sumber daya yang mewakili server Apache untuk me-restart server. Untuk informasi selengkapnya, lihat template.

Konfigurasi Stack dan Atribut Deployment: Linux

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Topik ini mencakup konfigurasi tumpukan dan atribut penerapan yang paling umum digunakan dan sintaks node yang terkait. Ini diatur di sekitar struktur namespace konfigurasi tumpukan yang digunakan oleh tumpukan Linux. Perhatikan bahwa nama atribut yang sama terkadang digunakan untuk tujuan yang berbeda, dan terjadi di ruang nama yang berbeda. Misalnya, `id` dapat merujuk ke ID tumpukan, ID lapisan, ID aplikasi, dan sebagainya, sehingga Anda memerlukan nama yang sepenuhnya memenuhi syarat untuk menggunakan nilai atribut. Cara mudah untuk memvisualisasikan data ini adalah sebagai objek JSON. Sebagai contoh, lihatKonfigurasi Stack dan Atribut Deployment.

> **Note**
>
> Pada instance Linux, AWS OpsWorks Stacks menginstal objek JSON ini pada setiap instance selain menambahkan data ke objek node. Anda dapat mengambilnya dengan menggunakan perintah agen CLI. `get_json`

Topik

- Atribut opsworks
- opsworks_custom_cookbooks Atribut
- dependensi Atribut
- Atribut ganglia
- Atribut mysql
- Atribut penumpang
- opsworks_bundler Atribut
- menyebarkan Atribut

- Atribut Tingkat Atas Lainnya

Atribut opsworks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

`opsworks`Elemen—kadang-kadang disebut sebagai `opsworks` namespace—berisi sekumpulan atribut yang menentukan konfigurasi tumpukan dasar.

> **Important**
>
> Mengganti nilai atribut di `opsworks` namespace tidak disarankan. Melakukannya dapat menyebabkan resep bawaan gagal.

Topik

- aplikasi
- Atribut contoh
- lapisan Atribut
- Atribut rails_stack
- Atribut tumpukan
- Atribut opsworks tingkat atas lainnya

aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan

pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Berisi daftar objek yang disematkan, satu untuk setiap aplikasi yang ada untuk tumpukan. Setiap objek tertanam berisi atribut berikut yang menggambarkan konfigurasi aplikasi.

Note

Sintaks node umum untuk atribut ini adalah sebagai berikut, di mana *i* menentukan indeks daftar berbasis nol instance.

```
node["opsworks"]["applications"]["i"]["attribute_name"]
```

application_type

Jenis aplikasi (string). Kemungkinan nilainya adalah sebagai berikut:

- `php`: Aplikasi PHP
- `rails`: Aplikasi Ruby on Rails
- `java`: Aplikasi Java
- `nodejs`: Aplikasi Node.js
- `web`: Halaman HTML statis
- `other`: Semua jenis aplikasi lainnya

```
node["opsworks"]["applications"]["i"]["application_type"]
```

name

Nama tampilan yang ditentukan pengguna, seperti `"SimplePHP"` (string).

```
node["opsworks"]["applications"]["i"]["name"]
```

slug_nama

Nama pendek, yang merupakan nama huruf kecil seperti `"simplephp"` yang dihasilkan oleh OpsWorks dari nama aplikasi (string).

```
node["opsworks"]["applications"]["i"]["slug_name"]
```

Atribut contoh

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

`instance`Atribut berisi satu set atribut yang menentukan konfigurasi instance ini.

arsitektur	availability_zone	backend
aws_instance_id	hostname	id
instance_type	ip	lapisan
private_dns_name	private_ip	public_dns_name
region		

arsitektur

Arsitektur instance, seperti `"i386"` (string).

```
node["opsworks"]["instance"]["architecture"]
```

availability_zone

Zona ketersediaan instance, seperti `"us-west-2a"` (string).

```
node["opsworks"]["instance"]["availability_zone"]
```

backend

Jumlah proses web back-end (string). Ini menentukan, misalnya, jumlah koneksi bersamaan yang HAProxy akan diteruskan ke ujung belakang Rails. Nilai default tergantung pada memori instance dan jumlah core.

```
node["opsworks"]["instance"]["backends"]
```

aws_instance_id

EC2 Contoh ID (string).

```
node["opsworks"]["instance"]["aws_instance_id"]
```

hostname

Nama host, seperti `"php-app1"` (string).

```
node["opsworks"]["instance"]["hostname"]
```

id

ID instance, yang merupakan GUID yang AWS OpsWorks dihasilkan Stacks yang secara unik mengidentifikasi instance (string).

```
node["opsworks"]["instance"]["id"]
```

instance_type

Jenis instance, seperti `"c1.medium"` (string).

```
node["opsworks"]["instance"]["instance_type"]
```

ip

Alamat IP publik (string).

```
node["opsworks"]["instance"]["ip"]
```

lapisan

Daftar lapisan instance, yang diidentifikasi dengan nama pendeknya, seperti `"lb"` atau `"db-master"` (daftar string).

```
node["opsworks"]["instance"]["layers"]
```

private_dns_name

Nama DNS pribadi (string).

```
node["opsworks"]["instance"]["private_dns_name"]
```

private_ip

Alamat IP pribadi (string).

```
node["opsworks"]["instance"]["private_ip"]
```

public_dns_name

Nama DNS publik (string).

```
node["opsworks"]["instance"]["public_dns_name"]
```

region

Wilayah AWS, seperti `"us-west-2"` (string).

```
node["opsworks"]["instance"]["region"]
```

lapisan Atribut

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

> Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

`layers`Atribut berisi satu set atribut lapisan, satu untuk setiap lapisan tumpukan, yang diberi nama dengan nama pendek lapisan, sepertiphp-app. Sebuah tumpukan dapat memiliki paling banyak satu setiap lapisan bawaan, yang nama pendeknya adalah sebagai berikut:

- `db-master`: Lapisan MySQL
- `java-app`: Lapisan Server Aplikasi Java
- `lb`: HAProxy lapisan
- `monitoring-master`: Lapisan ganglia
- `memcached`: Lapisan memcached
- `nodejs-app`: Lapisan Server Aplikasi Node.js
- `php-app`: Lapisan Server Aplikasi PHP
- `rails-app`: Lapisan Server Aplikasi Rails
- `web`: Lapisan Server Web Statis

Tumpukan dapat berisi sejumlah lapisan khusus, yang memiliki nama pendek yang ditentukan pengguna.

Setiap atribut layer berisi atribut berikut:

- id
- contoh
- name

id

> Layer ID, yang merupakan GUID yang dihasilkan oleh OpsWorks dan secara unik mengidentifikasi layer (string).

```
node["opsworks"]["layers"]["layershortname"]["id"]
```

Instans

`instances`Elemen berisi satu set atribut instance, satu untuk setiap instance online layer. Mereka diberi nama dengan nama host instance, sepertiphp-app1.

> **Note**
>
> `instances`Elemen hanya berisi instance yang berada dalam status online saat konfigurasi tumpukan dan atribut penerapan tertentu dibuat.

Setiap elemen instance berisi atribut berikut:

availability_zone	aws_instance_id	backend
booted_at	dibuat_at	elastic_ip
instance_type	ip	private_ip
public_dns_name	private_dns_name	region
status		

availability_zone

Availability Zone, seperti `"us-west-2a"` (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["availability_zone"]
```

aws_instance_id

EC2 Contoh ID (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["aws_instance_id"]
```

backend

Jumlah proses web back-end (nomor). Ini menentukan, misalnya, jumlah koneksi bersamaan yang HAProxy akan diteruskan ke ujung belakang Rails. Nilai default tergantung pada memori instance dan jumlah core.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["backends"]
```

booted_at

Waktu EC2 instance di-boot, menggunakan format UTC:mm:SS+HH:mm yyyy-mm-dddThh (string). Misalnya, `"2013-10-01T08:35:22+00:00"` sesuai dengan 8:35:22 pada 10 Oktober 2013, tanpa offset zona waktu. Untuk informasi selengkapnya, lihat ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["booted_at"]
```

dibuat_at

Waktu EC2 instance dibuat, menggunakan format UTC:mm:SS+HH:mm yyyy-mm-dddThh (string). Misalnya, `"2013-10-01T08:35:22+00:00"` sesuai dengan 8:35:22 pada 10 Oktober 2013, tanpa offset zona waktu. Untuk informasi selengkapnya, lihat ISO 8601.

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["created_at"]
```

elastic_ip

Alamat IP Elastis, yang diatur ke null jika instance tidak memiliki satu (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["elastic_ip"]
```

instance_type

Jenis instance, seperti `"c1.medium"` (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["instance_type"]
```

ip

Alamat IP publik (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["ip"]
```

private_ip

Alamat IP pribadi (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_ip"]
```

public_dns_name

Nama DNS publik (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["public_dns_name"]
```

private_dns_name

Nama DNS pribadi (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["private_dns_name"]
```

region

Wilayah AWS, seperti `"us-west-2"` (string).

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["region"]
```

status

Status (string). Kemungkinan nilainya adalah sebagai berikut:

- `"requested"`

- "booting"
- "running_setup"
- "online"
- "setup_failed"
- "start_failed"
- "terminating"
- "terminated"
- "stopped"
- "connection_lost"

```
node["opsworks"]["layers"]["layershortname"]["instances"]["instancehostname"]
["status"]
```

name

Nama layer, yang digunakan untuk mewakili lapisan di konsol (string). Itu bisa ditentukan pengguna dan belum tentu unik.

```
node["opsworks"]["layers"]["layershortname"]["name"]
```

Atribut rails_stack

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

name

Menentukan tumpukan rel, dan diatur ke "apache_passenger" atau "nginx_unicorn" (string).

```
node["opsworks"]["rails_stack"]["name"]
```

resep

Resep terkait, yang tergantung pada apakah Anda menggunakan Penumpang atau Unicorn (string):

- Unicorn: `"unicorn::rails"`
- Penumpang: `"passenger_apache2::rails"`

```
node["opsworks"]["rails_stack"]["recipe"]
```

restart_command

Perintah restart, yang tergantung pada apakah Anda menggunakan Passenger atau Unicorn (string):

- Unicorn: `"../../shared/scripts/unicorn clean-restart"`
- Penumpang: `"touch tmp/restart.txt"`

layanan

Nama layanan, yang tergantung pada apakah Anda menggunakan Passenger atau Unicorn (string):

- Unicorn: `"unicorn"`
- Penumpang: `"apache2"`

```
node["opsworks"]["rails_stack"]["service"]
```

Atribut tumpukan

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

`stack`atribut menentukan beberapa aspek konfigurasi tumpukan, seperti konfigurasi lapisan layanan.

- elb-load-balancers
- id
- name
- rds_instance
- vpc_id

elb-load-balancers

Berisi daftar objek yang disematkan, satu untuk setiap penyeimbang beban Elastic Load Balancing di tumpukan. Setiap objek tertanam berisi atribut berikut yang menggambarkan konfigurasi penyeimbang beban.

> **Note**
>
> Sintaks node umum untuk atribut ini adalah sebagai berikut, di mana *i* menentukan indeks daftar berbasis nol instance.
>
> ```
> node["opsworks"]["stack"]["elb-load-balancers"]["i"]["attribute_name"]
> ```

dns_nama

Nama DNS load balancer (string).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["dns_name"]
```

name

Nama load balancer (string).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["name"]
```

layer_id

ID lapisan tempat penyeimbang beban dilampirkan (string).

```
node["opsworks"]["stack"]["elb-load-balancers"]["i"]["layer_id"]
```

id

ID tumpukan (string).

```
node["opsworks"]["stack"]["id"]
```

name

Nama tumpukan (string).

```
node["opsworks"]["stack"]["name"]
```

rds_instance

Berisi daftar objek yang disematkan, satu untuk setiap instans Amazon RDS yang terdaftar dengan tumpukan. Setiap objek tertanam berisi satu set atribut yang menentukan konfigurasi instance. Anda menentukan nilai ini saat menggunakan konsol Amazon RDS atau API untuk membuat instance. Anda juga dapat menggunakan konsol Amazon RDS atau API untuk mengedit beberapa pengaturan setelah instance dibuat. Untuk informasi selengkapnya, lihat dokumentasi Amazon RDS.

> **Note**
>
> Sintaks node umum untuk atribut ini adalah sebagai berikut, di mana *i* menentukan indeks daftar berbasis nol instance.
>
> ```
> node["opsworks"]["stack"]["rds_instances"]["i"]["attribute_name"]
> ```

Jika tumpukan Anda memiliki beberapa instans Amazon RDS, berikut ini adalah contoh cara menggunakan instance tertentu dalam resep.

```
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
```

alamat	allocated_storage	arn
auto_minor_version_upgrade	availability_zone	backup_retention_period
db_instance_class	db_instance_identifier	db_instance_status
db_nama	db_parameter_groups	db_security_groups
db_pengguna	engine	instance_create_time
license_model	multi_az	option_group_memberships
port	disukai_backup_window	preferred_maintenance_window
publicly_accessible	read_replica_db_instance_identifiers	region
status_infos	vpc_security_groups	

alamat

Contoh URL, seperti `opsinstance.ccdvt3hwog1a.us-west-2.rds.amazonaws.com` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["address"]
```

allocated_storage

Penyimpanan yang dialokasikan, dalam GB (angka).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["allocated_storage"]
```

arn

Contohnya ARN (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["arn"]
```

auto_minor_version_upgrade

Apakah akan secara otomatis menerapkan upgrade versi minor (Boolean).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["auto_minor_version_upgrade"]
```

availability_zone

Availability Zone instance, seperti `us-west-2a` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["availability_zone"]
```

backup_retention_period

Periode retensi cadangan, dalam beberapa hari (angka).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["backup_retention_period"]
```

db_instance_class

Kelas instance DB, seperti `db.m1.small` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_class"]
```

db_instance_identifier

Pengidentifikasi instans DB yang ditentukan pengguna (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_identifier"]
```

db_instance_status

Status instance (string). Untuk informasi selengkapnya, lihat Instans DB.

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_instance_status"]
```

db_nama

Nama DB yang ditentukan pengguna (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_name"]
```

db_parameter_groups

Grup parameter DB instance, yang berisi daftar objek yang disematkan, satu untuk setiap grup parameter. Untuk informasi selengkapnya, lihat Bekerja dengan Grup Parameter DB. Setiap objek berisi atribut berikut:

db_parameter_group_name

Nama grup (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["db_parameter_group_name"]
```

parameter_apply_status

Status terapkan (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_parameter_groups"][j"]
["parameter_apply_status"]
```

db_security_groups

Grup keamanan database instance, yang berisi daftar objek tertanam, satu untuk setiap grup keamanan. Untuk informasi selengkapnya, lihat Bekerja dengan Grup Keamanan DB. Setiap objek berisi atribut berikut

db_security_group_name

Nama grup keamanan (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["db_security_group_name"]
```

status

Status (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_security_groups"][j"]
["status"]
```

db_pengguna

Nama Pengguna Master yang ditentukan pengguna (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["db_user"]
```

engine

Mesin database, seperti `mysql(5.6.13)` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["engine"]
```

instance_create_time

Waktu pembuatan instance, seperti `2014-04-15T16:13:34Z` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["instance_create_time"]
```

license_model

Model lisensi instance, seperti `general-public-license` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["license_model"]
```

multi_az

Apakah penerapan Multi-AZ diaktifkan (Boolean).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["multi_az"]
```

option_group_memberships

Keanggotaan grup opsi instance, yang berisi daftar objek yang disematkan, satu untuk setiap grup opsi. Untuk informasi selengkapnya, lihat Bekerja dengan Grup Opsi. Setiap objek berisi atribut berikut:

option_group_name

Nama grup (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["option_group_name"]
```

status

Status grup (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["option_group_memberships"]
[j"]["status"]
```

port

Port server database (nomor).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["port"]
```

disukai_backup_window

Jendela cadangan harian yang disukai, seperti `06:26-06:56` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_backup_window"]
```

preferred_maintenance_window

Jendela pemeliharaan mingguan yang disukai, seperti `thu:07:13-thu:07:43` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["preferred_maintenance_window"]
```

publicly_accessible

Apakah database dapat diakses publik (Boolean).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["publicly_accessible"]
```

read_replica_db_instance_identifiers

Daftar pengidentifikasi instance read-replica (daftar string). Untuk informasi selengkapnya, lihat Bekerja dengan Read Replicas.

```
node["opsworks"]["stack"]["rds_instances"]["i"]
["read_replica_db_instance_identifiers"]
```

region

Wilayah AWS, seperti `us-west-2` (string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["region"]
```

status_infos

Daftar informasi status (daftar string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["status_infos"]
```

vpc_security_groups

Daftar grup keamanan VPC (daftar string).

```
node["opsworks"]["stack"]["rds_instances"]["i"]["vpc_security_groups"]
```

vpc_id

Id VPC (string). Nilai ini adalah `null` jika instance tidak dalam VPC.

```
node["opsworks"]["stack"]["vpc_id"]
```

Atribut opsworks tingkat atas lainnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini berisi `opsworks` atribut yang tidak memiliki atribut anak.

aktivitas

Aktivitas yang terkait dengan atribut, seperti `deploy` (string).

```
node["opsworks"]["activity"]
```

agent_version

Versi OpsWorks agen instance (string).

```
node["opsworks"]["agent_version"]
```

deploy_chef_provider

Penyedia penerapan Chef, yang memengaruhi struktur direktori (string) aplikasi yang diterapkan. Anda dapat mengatur atribut ini ke salah satu berikut:

- `Branch`
- `Revision`
- `Timestamped`(nilai default)

```
node["opsworks"]["deploy_chef_provider"]
```

ruby_stack

Tumpukan Ruby (string). Pengaturan default adalah versi perusahaan (`ruby_enterprise`). Untuk versi MRI, atur atribut ini ke`ruby`.

```
node["opsworks"]["ruby_stack"]
```

ruby_version

Versi Ruby yang akan digunakan oleh aplikasi (string). Anda dapat menggunakan atribut ini untuk menentukan hanya versi mayor dan minor. Anda harus menggunakan atribut ["ruby"] yang sesuai untuk menentukan versi tambalan. Untuk informasi selengkapnya tentang cara menentukan versi, termasuk contoh, lihatVersi Ruby. Untuk detail lengkap tentang cara AWS OpsWorks Stacks menentukan versi Ruby, lihat file atribut bawaan, ruby.rb.

```
node["opsworks"]["ruby_version"]
```

run_cookbook_tests

Apakah akan menjalankan minitest-chef-handlertes pada buku masak Chef 11.4 Anda (Boolean).

```
node["opsworks"]["run_cookbook_tests"]
```

sent_at

Ketika perintah ini dikirim ke instance (nomor).

```
node["opsworks"]["sent_at"]
```

deployment

Jika atribut ini dikaitkan dengan aktivitas penerapan, `deployment` disetel ke ID penerapan, GUID AWS OpsWorks buatan Stacks yang secara unik mengidentifikasi penerapan (string). Jika tidak, atribut diatur ke null.

```
node["opsworks"]["deployment"]
```

opsworks_custom_cookbooks Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Berisi atribut yang menentukan buku masak khusus tumpukan.

diaktifkan

Apakah buku masak khusus diaktifkan (Boolean).

```
node["opsworks_custom_cookbooks"]["enabled"]
```

resep

Daftar resep yang akan dijalankan untuk perintah ini, termasuk resep khusus, menggunakan *`cookbookname`*`::`*`recipename`* format (daftar string).

```
node["opsworks_custom_cookbooks"]["recipes"]
```

dependensi Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Berisi beberapa atribut yang terkait dengan perintah `update_dependencies` stack.

gem_binary

Lokasi biner Permata (string).

upgrade_debs

Apakah akan memutakhirkan paket Debs (Boolean).

update_debs

Apakah akan memperbarui paket Debs (Boolean).

Atribut ganglia

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Berisi atribut web yang berisi beberapa atribut yang menentukan cara mengakses halaman web statistik Ganglia:

password

Kata sandi yang diperlukan untuk mengakses halaman statistik (string).

```
node["ganglia"]["web"]["password"]
```

url

Jalur URL halaman statistik, seperti `"/ganglia"` (string). URL lengkapnya adalah`http://`*`DNSNameURLPath`*, di mana *`DNSName`* nama DNS instance terkait.

```
node["ganglia"]["web"]["url"]
```

user

Nama pengguna diperlukan untuk mengakses halaman statistik (string).

```
node["ganglia"]["web"]["user"]
```

Atribut mysql

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Berisi satu set atribut yang menentukan konfigurasi server database MySQL.

klien

Daftar alamat IP klien (daftar string).

```
node["mysql"]["clients"]
```

server_root_password

Kata sandi root (string).

```
node["mysql"]["server_root_password"]
```

Atribut penumpang

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Berisi satu set atribut yang menentukan konfigurasi Phusion Passenger.

gem_bin

Lokasi RubyGems binari, seperti "`/usr/local/bin/gem`" (string).

```
node["passenger"]["gem_bin"]
```

max_pool_size

Ukuran kolam maksimum (jumlah).

```
node["passenger"]["max_pool_size"]
```

ruby_bin

Lokasi binari Ruby, seperti. "`/usr/local/bin/ruby`"

```
node["passenger"]["ruby_bin"]
```

versi

Versi Penumpang (string).

```
node["passenger"]["version"]
```

opsworks_bundler Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Berisi elemen yang menentukan dukungan Bundler.

manage_package

Apakah akan menginstal dan mengelola Bundler (Boolean).

```
node["opsworks_bundler"]["manage_package"]
```

versi

Versi bundler (string).

```
node["opsworks_bundler"]["version"]
```

menyebarkan Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika atribut dikaitkan dengan peristiwa Deploy atau perintah stack Execute Recipes, `deploy` atribut tersebut berisi atribut untuk setiap aplikasi yang di-deploy, dinamai dengan nama pendek aplikasi. Setiap atribut app berisi atribut berikut:

application	application_type	auto_bundle_on_deploy
basis data	deploy_to	wilayah
document_root	environment_variable	grup
keep_releases	memcached	bermigrasi
mounted_at	purge_before_symlink	rails_env
restart_command	scm	ssl_certificate
ssl_certificate_ca	ssl_certificate_key	ssl_support
tumpukan	symlink_before_migrate	symlink
user		

application

Nama slug aplikasi, seperti `"simplephp"` (string).

```
node["deploy"]["appshortname"]["application"]
```

application_type

Jenis aplikasi (string). Kemungkinan nilainya adalah sebagai berikut:

- `java`: Aplikasi Java
- `nodejs`: Aplikasi Node.js
- `php`: Aplikasi PHP
- `rails`: Aplikasi Ruby on Rails
- `web`: Halaman HTML statis
- `other`: Semua jenis aplikasi lainnya

```
node["deploy"]["appshortname"]["application_type"]
```

auto_bundle_on_deploy

Untuk aplikasi Rails, apakah akan mengeksekusi bundler selama penerapan (Boolean).

```
node["deploy"]["appshortname"]["auto_bundle_on_deploy"]
```

basis data

Berisi informasi yang diperlukan untuk menghubungkan database aplikasi. Jika aplikasi memiliki lapisan database terlampir, AWS OpsWorks Stacks secara otomatis menetapkan nilai yang sesuai untuk atribut ini.

adaptor

Adaptor database, seperti `mysql` (string).

```
node["deploy"]["appshortname"]["database"]["adapter"]
```

basis data

Nama database, yang biasanya merupakan nama slug aplikasi, seperti `"simplephp"` (string).

```
node["deploy"]["appshortname"]["database"]["database"]
```

data_source_provider

Sumber data: `mysql` atau `rds` (string).

```
node["deploy"]["appshortname"]["database"]["data_source_provider"]
```

host

Alamat IP host database (string).

```
node["deploy"]["appshortname"]["database"]["host"]
```

password

Kata sandi basis data (string).

```
node["deploy"]["appshortname"]["database"]["password"]
```

port

Port database (nomor).

```
node["deploy"]["appshortname"]["database"]["port"]
```

sambungkan kembali

Untuk aplikasi Rails, apakah aplikasi harus menyambung kembali jika koneksi tidak ada lagi (Boolean).

```
node["deploy"]["appshortname"]["database"]["reconnect"]
```

nama pengguna

Nama pengguna (string).

```
node["deploy"]["appshortname"]["database"]["username"]
```

deploy_to

Di mana aplikasi akan digunakan, seperti `"/srv/www/simplephp"` (string).

```
node["deploy"]["appshortname"]["deploy_to"]
```

wilayah

Daftar domain aplikasi (daftar string).

```
node["deploy"]["appshortname"]["domains"]
```

document_root

Root dokumen, jika Anda menentukan root nondefault, atau null jika Anda menggunakan root default (string).

```
node["deploy"]["appshortname"]["document_root"]
```

environment_variable

Kumpulan hingga dua puluh atribut yang mewakili variabel lingkungan yang ditentukan pengguna yang telah ditentukan untuk aplikasi. Untuk informasi selengkapnya tentang cara menentukan variabel lingkungan aplikasi, lihatMenambahkan Aplikasi. Setiap nama atribut diatur ke nama variabel lingkungan dan nilai yang sesuai diatur ke nilai variabel, sehingga Anda dapat menggunakan sintaks berikut untuk referensi nilai tertentu.

```
node["deploy"]["appshortname"]["environment_variables"]["variable_name"]
```

grup

Grup aplikasi (string).

```
node["deploy"]["appshortname"]["group"]
```

keep_releases

Jumlah penerapan aplikasi yang akan disimpan AWS OpsWorks Stacks (nomor). Atribut ini mengontrol berapa kali Anda dapat memutar kembali aplikasi. Secara default, ini diatur ke nilai global,deploy_keep_releases , yang memiliki nilai default 5. Anda dapat mengganti `keep_releases` untuk menentukan jumlah penyebaran tersimpan untuk aplikasi tertentu.

```
node["deploy"]["appshortname"]["keep_releases"]
```

memcached

Berisi dua atribut yang menentukan konfigurasi memcached.

host

Alamat IP (string) instance server Memcached.

```
node["deploy"]["appshortname"]["memcached"]["host"]
```

port

Port yang didengarkan server memcached (nomor).

```
node["deploy"]["appshortname"]["memcached"]["port"]
```

bermigrasi

Untuk aplikasi Rails, apakah akan menjalankan migrasi (Boolean).

```
node["deploy"]["appshortname"]["migrate"]
```

mounted_at

Titik pemasangan aplikasi, jika Anda menentukan titik pemasangan nondefault, atau null jika Anda menggunakan titik pemasangan (string) default.

```
node["deploy"]["appshortname"]["mounted_at"]
```

purge_before_symlink

Untuk aplikasi Rails, larik jalur yang akan dihapus sebelum membuat symlink (daftar string).

```
node["deploy"]["appshortname"]["purge_before_symlink"]
```

rails_env

Untuk instance Rails App Server, lingkungan Rails, seperti `"production"` (string).

```
node["deploy"]["appshortname"]["rails_env"]
```

restart_command

Perintah yang akan dijalankan saat aplikasi dimulai ulang, seperti. `"echo 'restarting app'"`

```
node["deploy"]["appshortname"]["restart_command"]
```

scm

Berisi sekumpulan atribut yang menentukan informasi yang OpsWorks digunakan untuk menyebarkan aplikasi dari repositori kontrol sumbernya. Atribut bervariasi tergantung pada jenis repositori.

password

Kata sandi, untuk repositori pribadi, dan null untuk repositori publik (string). Untuk bucket Amazon S3 pribadi, atribut diatur ke kunci rahasia.

```
node["deploy"]["appshortname"]["scm"]["password"]
```

repositori

URL repositori, seperti `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (string).

```
node["deploy"]["appshortname"]["scm"]["repository"]
```

revisi

Jika repositori memiliki beberapa cabang, atribut menentukan cabang atau versi aplikasi, seperti `"version1"` (string). Jika tidak, itu diatur ke null.

```
node["deploy"]["appshortname"]["scm"]["revision"]
```

scm_type

Jenis repositori (string). Kemungkinan nilainya adalah sebagai berikut:

- `"git"`: Sebuah repositori Git
- `"svn"`: Sebuah repositori Subversion
- `"s3"`: Ember Amazon S3
- `"archive"`: Sebuah arsip HTTP
- `"other"`: Jenis repositori lain

```
node["deploy"]["appshortname"]["scm"]["scm_type"]
```

ssh_key

Kunci SSH deploy, untuk mengakses repositori Git pribadi, dan null untuk repositori publik (string).

```
node["deploy"]["appshortname"]["scm"]["ssh_key"]
```

user

Nama pengguna, untuk repositori pribadi, dan null untuk repositori publik (string). Untuk bucket Amazon S3 pribadi, atribut diatur ke kunci akses.

```
node["deploy"]["appshortname"]["scm"]["user"]
```

ssl_certificate

Sertifikat SSL aplikasi, jika Anda mengaktifkan dukungan SSL, atau null sebaliknya (string).

```
node["deploy"]["appshortname"]["ssl_certificate"]
```

ssl_certificate_ca

Jika SSL diaktifkan, atribut untuk menentukan kunci otoritas sertifikat menengah atau otentikasi klien (string).

```
node["deploy"]["appshortname"]["ssl_certificate_ca"]
```

ssl_certificate_key

Kunci pribadi SSL aplikasi, jika Anda mengaktifkan dukungan SSL, atau null sebaliknya (string).

```
node["deploy"]["appshortname"]["ssl_certificate_key"]
```

ssl_support

Apakah SSL didukung (Boolean).

```
node["deploy"]["appshortname"]["ssl_support"]
```

tumpukan

Berisi satu atribut Boolean,`needs_reload`, yang menentukan apakah akan memuat ulang server aplikasi selama penerapan.

```
node["deploy"]["appshortname"]["stack"]["needs_reload"]
```

symlink_before_migrate

Untuk aplikasi Rails, berisi symlink yang akan dibuat sebelum menjalankan migrasi sebagai pasangan. `"link":"target"`

```
node["deploy"]["appshortname"]["symlink_before_migrate"]
```

symlink

Berisi symlink penerapan sebagai "*link*":"*target*" pasangan.

```
node["deploy"]["appshortname"]["symlinks"]
```

user

Pengguna aplikasi (string).

```
node["deploy"]["appshortname"]["user"]
```

Atribut Tingkat Atas Lainnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini berisi atribut konfigurasi tumpukan tingkat atas yang tidak memiliki atribut anak.

Atribut rel

Berisi atribut max_pool_size yang menentukan ukuran kolam maksimum server (nomor). Nilai atribut diatur oleh AWS OpsWorks Stacks dan bergantung pada jenis instance, tetapi Anda dapat menggantinya dengan menggunakan JSON kustom atau file atribut kustom.

```
node["rails"]["max_pool_size"]
```

resep Atribut

Daftar resep bawaan yang dijalankan oleh aktivitas ini, menggunakan "*cookbookname*::*recipename*" format (daftar string).

```
node["recipes"]
```

opsworks_rubygems Atribut

Berisi elemen versi yang menentukan RubyGems versi (string).

```
node["opsworks_rubygems"]["version"]
```

bahasa Atribut

Berisi atribut untuk setiap bahasa yang diinstal, dinamai untuk bahasa, seperti ruby. Atribut adalah objek yang berisi atribut, seperti ruby_bin, yang menentukan folder instalasi, seperti `"/usr/bin/ruby"` (string).

ssh_users Atribut

Berisi satu set atribut, masing-masing menggambarkan salah satu pengguna yang telah diberikan izin SSH. Setiap atribut diberi nama dengan ID Unix pengguna. AWS OpsWorks Stacks menghasilkan ID unik untuk setiap pengguna dalam rentang 2000-4000, seperti`"2001"`, dan membuat pengguna dengan ID tersebut di setiap instance. Karena AWS OpsWorks memiliki rentang 2000-4000, pengguna yang Anda buat di luar AWS OpsWorks (dengan menggunakan resep buku masak, atau dengan mengimpor pengguna ke AWS OpsWorks dari IAM, misalnya) dapat memiliki UIDs yang ditimpa oleh Tumpukan untuk pengguna lain. AWS OpsWorks Sebagai praktik terbaik, buat pengguna dan kelola akses mereka di konsol AWS OpsWorks Stacks. Jika Anda membuat pengguna di luar AWS OpsWorks Stacks, gunakan *`UnixID`* nilai yang lebih besar dari 4000.

Setiap atribut berisi atribut berikut:

Email

Alamat e-mail pengguna (string).

```
node["ssh_users"]["UnixID"]["email"]
```

public_key

Kunci SSH publik pengguna (string).

```
node["ssh_users"]["UnixID"]["public_key"]
```

sudoer

Apakah pengguna memiliki izin sudo (Boolean).

```
node["ssh_users"]["UnixID"]["sudoer"]
```

name

Nama pengguna (string).

```
node["ssh_users"]["UnixID"]["name"]
```

Atribut Buku Masak Bawaan

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Sebagian besar atribut ini hanya tersedia di tumpukan Linux.

Sebagian besar resep bawaan memiliki satu atau lebih file atribut yang menentukan berbagai pengaturan. Anda dapat mengakses pengaturan ini dalam resep khusus Anda dan menggunakan JSON khusus untuk menggantinya. Anda biasanya perlu mengakses atau mengganti atribut yang mengontrol konfigurasi berbagai teknologi server yang didukung oleh AWS OpsWorks Stacks. Bagian ini merangkum atribut-atribut tersebut. File atribut lengkap, serta resep dan templat terkait, tersedia di https://github.com/aws/opsworks-cookbooks.git.

Note

Semua atribut resep bawaan adalah `default` tipe.

Topik

- Atribut apache2
- menyebarkan Atribut
- Atribut haproxy
- Atribut memcached
- Atribut mysql
- nginx Atribut
- opsworks_berkshelf Atribut
- opsworks_java Atribut
- Atribut passenger_apache2
- Atribut ruby
- Atribut unicorn

Atribut apache2

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Atribut ini hanya tersedia di tumpukan Linux.

Atribut apache2 menentukan konfigurasi server HTTP Apache. Untuk informasi selengkapnya, lihat Fitur Inti Apache. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

biner	kontak	deflate_types

dir	document_root	grup
hide_info_headers	icondir	init_script
keepalive	keepaliverequests	keepalivetimeout
lib_dir	libexecdir	listen_ports
log_dir	Atribut logrotate	pid_file
Atribut prefork	tanda tangan server	servertoken
batas waktu	dapat dilacak	user
versi	Atribut pekerja	

biner

Lokasi biner Apache (string). Nilai default-nya adalah `'/usr/sbin/httpd'`.

```
node[:apache][:binary]
```

kontak

Kontak email (string). Nilai default adalah alamat dummy, `'ops@example.com'`.

```
node[:apache][:contact]
```

deflate_types

Mengarahkan `mod_deflate` untuk mengaktifkan kompresi untuk jenis Mime yang ditentukan, jika didukung oleh browser (daftar string). Nilai defaultnya adalah sebagai berikut:

```
['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**
>
> Kompresi dapat menimbulkan risiko keamanan. Untuk sepenuhnya menonaktifkan kompresi, atur atribut ini sebagai berikut:
>
> ```
> node[:apache][:deflate_types] = []
> ```

```
node[:apache][:deflate_types]
```

dir

Direktori root server (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan Red Hat Enterprise Linux (RHEL): `'/etc/httpd'`
- Ubuntu: `'/etc/apache2'`

```
node[:apache][:dir]
```

document_root

Root dokumen (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'/var/www/html'`
- Ubuntu: `'/var/www'`

```
node[:apache][:document_root]
```

grup

Nama grup (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'apache'`
- Ubuntu: `'www-data'`

```
node[:apache][:group]
```

hide_info_headers

Apakah akan menghilangkan informasi versi dan modul dari header HTTP (`'true'`/`'false'`) (string). Nilai default-nya adalah `'true'`.

```
node[:apache][:hide_info_headers]
```

icondir

Direktori ikon (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'/var/www/icons/'`
- Ubuntu: `'/usr/share/apache2/icons'`

```
node[:apache][:icondir]
```

init_script

Script inisialisasi (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'/etc/init.d/httpd'`
- Ubuntu: `'/etc/init.d/apache2'`

```
node[:apache][:init_script]
```

keepalive

Apakah akan mengaktifkan koneksi keep-alive (string). Nilai yang mungkin adalah `'On'` dan `'Off'` (string). Nilai default-nya adalah `'Off'`.

```
node[:apache][:keepalive]
```

keepaliverequests

Jumlah maksimum permintaan keep-alive yang akan ditangani Apache pada saat yang sama (nomor). Nilai default-nya adalah `100`.

```
node[:apache][:keepaliverequests]
```

keepalivetimeout

Waktu Apache menunggu permintaan sebelum menutup koneksi (nomor). Nilai default-nya adalah `3`.

```
node[:apache][:keepalivetimeout]
```

lib_dir

Direktori yang berisi pustaka kode objek (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux (x86): `'/usr/lib/httpd'`
- Amazon Linux (x64) dan RHEL: `'/usr/lib64/httpd'`
- Ubuntu: `'/usr/lib/apache2'`

```
node[:apache][:lib_dir]
```

libexecdir

Direktori yang berisi program executable (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux (x86): `'/usr/lib/httpd/modules'`
- Amazon Linux (x64) dan RHEL: `'/usr/lib64/httpd/modules'`
- Ubuntu: `'/usr/lib/apache2/modules'`

```
node[:apache][:libexecdir]
```

listen_ports

Daftar port yang didengarkan server (daftar string). Nilai default-nya adalah `[ '80','443' ]`.

```
node[:apache][:listen_ports]
```

log_dir

Direktori log (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'/var/log/httpd'`
- Ubuntu: `'/var/log/apache2'`

```
node[:apache][:log_dir]
```

Atribut logrotate

Atribut ini menentukan cara memutar file log.

penundaankompres

Apakah akan menunda mengompresi file log tertutup sampai dimulainya siklus rotasi berikutnya (`'true'`/`'false'`) (string). Nilai default-nya adalah `'true'`.

```
node[:apache][:logrotate][:delaycompress]
```

grup

Grup file log (string). Nilai default-nya adalah `'adm'`.

```
node[:apache][:logrotate][:group]
```

Mode

Mode file log (string). Nilai default-nya adalah `'640'`.

```
node[:apache][:logrotate][:mode]
```

owner

Pemilik file log (string). Nilai default-nya adalah `'root'`.

```
node[:apache][:logrotate][:owner]
```

merotasi

Jumlah siklus rotasi sebelum file log tertutup dihapus (string). Nilai default-nya adalah `'30'`.

```
node[:apache][:logrotate][:rotate]
```

jadwal

Jadwal rotasi (string). Kemungkinan nilainya adalah sebagai berikut:

- `'daily'`
- `'weekly'`
- `'monthly'`

Nilai default-nya adalah `'daily'`.

```
node[:apache][:logrotate][:schedule]
```

pid_file

File yang berisi ID proses daemon (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'/var/run/httpd/httpd.pid'`
- Ubuntu: `'/var/run/apache2.pid'`

```
node[:apache][:pid_file]
```

Atribut prefork

Atribut ini menentukan konfigurasi pra-forking.

maxclient

Jumlah maksimum permintaan simultan yang akan dilayani (nomor). Nilai default-nya adalah `400`.

> **Note**
>
> Gunakan atribut ini hanya untuk instance yang menjalankan Amazon Linux, atau RHEL. Jika instance Anda menjalankan Ubuntu 14.04 LTS, gunakan. maxrequestworkers

```
node[:apache][:prefork][:maxclients]
```

maxrequestsperchild

Jumlah maksimum permintaan yang akan ditangani oleh proses server anak (nomor). Nilai default-nya adalah `10000`.

```
node[:apache][:prefork][:maxrequestsperchild]
```

maxrequestworkers

Jumlah maksimum permintaan simultan yang akan dilayani (nomor). Nilai default-nya adalah `400`.

> **Note**
>
> Gunakan atribut ini hanya untuk instance yang menjalankan Ubuntu 14.04 LTS. Jika instans Anda menjalankan Amazon Linux, atau RHEL, gunakan. maxclient

```
node[:apache][:prefork][:maxrequestworkers]
```

server maxspareserver

Jumlah maksimum proses server anak idle (angka). Nilai default-nya adalah `32`.

```
node[:apache][:prefork][:maxspareservers]
```

server minspareserver

Jumlah minimum proses server anak idle (angka). Nilai default-nya adalah `16`.

```
node[:apache][:prefork][:minspareservers]
```

batas server

Jumlah maksimum proses yang dapat dikonfigurasi (angka). Nilai default-nya adalah `400`.

```
node[:apache][:prefork][:serverlimit]
```

server awal

Jumlah proses server anak yang akan dibuat saat startup (angka). Nilai default-nya adalah `16`.

```
node[:apache][:prefork][:startservers]
```

tanda tangan server

Menentukan apakah dan bagaimana mengkonfigurasi footer trailing untuk dokumen server-dihasilkan (string). Nilai yang mungkin adalah`'On'`,`'Off'`, dan`'Email'`). Nilai default-nya adalah `'Off'`.

```
node[:apache][:serversignature]
```

servertoken

Menentukan jenis informasi versi server termasuk dalam header respon (string):

- `'Full'`: Informasi lengkap. Misalnya, Server: Apache/2.4.2 (Unix) PHP/4.2.2 /1.2 MyMod
- `'Prod'`: Nama produk. Misalnya, Server: Apache
- `'Major'`: Versi utama. Misalnya, Server: Apache/2
- `'Minor'`: Versi mayor dan minor. Misalnya, Server: Apache/2.4
- `'Min'`: Versi minimal. Misalnya, Server: Apache/2.4.2
- `'OS'`: Versi dengan sistem operasi. Misalnya, Server: Apache/2.4.2 (Unix)

Nilai default-nya adalah `'Prod'`.

```
node[:apache][:servertokens]
```

batas waktu

Jumlah waktu yang Apache menunggu I/O (nomor). Nilai default-nya adalah `120`.

```
node[:apache][:timeout]
```

dapat dilacak

Apakah akan mengaktifkan `TRACE` permintaan (string). Nilai yang mungkin adalah `'On'` dan `'Off'`. Nilai default-nya adalah `'Off'`.

```
node[:apache][:traceenable]
```

user

Nama pengguna (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'apache'`
- Ubuntu: `'www-data'`

```
node[:apache][:user]
```

versi

Versi Apache (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux: `2.2`
- Ubuntu 14.04 LTS: `2.4`
- RHEL: `2.4`

```
node[:apache][:version]
```

Atribut pekerja

Atribut ini menentukan konfigurasi proses pekerja.

server awal

Jumlah proses server anak yang akan dibuat saat startup (angka). Nilai default-nya adalah `4`.

```
node[:apache][:worker][:startservers]
```

maxclient

Jumlah maksimum permintaan simultan yang akan dilayani (nomor). Nilai default-nya adalah `1024`.

```
node[:apache][:worker][:maxclients]
```

maxsparethreads

Jumlah maksimum utas idle (angka). Nilai default-nya adalah `192`.

```
node[:apache][:worker][:maxsparethreads]
```

minsparethreads

Jumlah minimum utas idle (angka). Nilai default-nya adalah `64`.

```
node[:apache][:worker][:minsparethreads]
```

threadsperchild

Jumlah utas per proses anak (nomor). Nilai default-nya adalah 64.

```
node[:apache][:worker][:threadsperchild]
```

maxrequestsperchild

Jumlah maksimum permintaan yang akan ditangani oleh proses server anak (nomor). Nilai default-nya adalah `10000`.

```
node[:apache][:worker][:maxrequestsperchild]
```

menyebarkan Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

File `deploy.rb` atribut buku masak deploy bawaan mendefinisikan atribut berikut di namespace. `opsworks` Untuk informasi selengkapnya tentang direktori penerapan, lihat. Menyebarkan Resep Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

deploy_keep_releases

Pengaturan global untuk jumlah penerapan aplikasi yang akan disimpan AWS OpsWorks Stacks (nomor). Nilai bawaannya adalah 5. Nilai ini mengontrol berapa kali Anda dapat memutar kembali aplikasi.

```
node[:opsworks][:deploy_keep_releases]
```

grup

(Hanya Linux) `group` Pengaturan untuk direktori penerapan aplikasi (string). Nilai default tergantung pada sistem operasi instans:

- Untuk contoh Ubuntu, nilai defaultnya adalah`www-data`.
- Untuk instans Amazon Linux atau RHEL yang merupakan anggota lapisan Rails App Server yang menggunakan Nginx dan Unicorn, nilai defaultnya adalah. `nginx`
- Untuk semua instans Amazon Linux atau RHEL lainnya, nilai defaultnya adalah. `apache`

```
node[:opsworks][:deploy_user][:group]
```

user

(Hanya Linux) `user` Pengaturan untuk direktori penerapan aplikasi (string). Nilai default-nya adalah `deploy`.

```
node[:opsworks][:deploy_user][:user]
```

Atribut haproxy

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Atribut ini hanya tersedia di tumpukan Linux.

`haproxyAtribut` menentukan konfigurasi HAProxy server. Untuk informasi selengkapnya, lihat HAProxyDokumen. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

keseimbangan	check_interval	client_timeout
connect_timeout	default_max_connections	global_max_connections
health_check_method	health_check_url	antrian_batas waktu
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
mencoba lagi	server_timeout	stats_url
stats_user		

keseimbangan

Algoritma yang digunakan oleh penyeimbang beban untuk memilih server (string). Nilai default-nya adalah `'roundrobin'`. Pilihan lainnya adalah:

- 'statis-rr'
- 'terkecil'
- 'sumber'
- 'uri'
- 'url_param'
- 'hdr (nama) '
- 'rdp-cookie'
- 'rdp-cookie (nama) '

Untuk informasi lebih lanjut tentang argumen ini, lihat keseimbangan.

```
node[:haproxy][:balance]
```

check_interval

Interval waktu pemeriksaan kesehatan (string). Nilai default-nya adalah `'10s'`.

```
node[:haproxy][:check_interval]
```

client_timeout

Jumlah maksimum waktu yang klien dapat tidak aktif (string). Nilai default-nya adalah `'60s'`.

```
node[:haproxy][:client_timeout]
```

connect_timeout

Jumlah maksimum waktu yang HAProxy akan menunggu upaya koneksi server untuk berhasil (string). Nilai default-nya adalah `'10s'`.

```
node[:haproxy][:connect_timeout]
```

default_max_connections

Jumlah maksimum default koneksi (string). Nilai default-nya adalah `'80000'`.

```
node[:haproxy][:default_max_connections]
```

global_max_connections

Jumlah maksimum koneksi (string). Nilai default-nya adalah `'80000'`.

```
node[:haproxy][:global_max_connections]
```

health_check_method

Metode pemeriksaan kesehatan (string). Nilai default-nya adalah `'OPTIONS'`.

```
node[:haproxy][:health_check_method]
```

health_check_url

Jalur URL yang digunakan untuk memeriksa kesehatan server (string). Nilai default-nya adalah `'/'`.

```
node[:haproxy][:health_check_url ]
```

antrian_batas waktu

Waktu tunggu maksimum untuk koneksi gratis (string). Nilai default-nya adalah `'120s'`.

```
node[:haproxy][:queue_timeout]
```

http_request_timeout

Jumlah maksimum waktu yang HAProxy akan menunggu permintaan HTTP lengkap (string). Nilai default-nya adalah `'30s'`.

```
node[:haproxy][:http_request_timeout]
```

mencoba lagi

Jumlah percobaan ulang setelah kegagalan koneksi server (string). Nilai default-nya adalah `'3'`.

```
node[:haproxy][:retries]
```

server_timeout

Jumlah maksimum waktu yang klien dapat tidak aktif (string). Nilai default-nya adalah `'60s'`.

```
node[:haproxy][:server_timeout]
```

stats_url

Jalur URL untuk halaman statistik (string). Nilai default-nya adalah `'/haproxy?stats'`.

```
node[:haproxy][:stats_url]
```

stats_user

Nama pengguna halaman statistik (string). Nilai default-nya adalah `'opsworks'`.

```
node[:haproxy][:stats_user]
```

`maxcon`Atribut mewakili pengganda faktor beban yang digunakan untuk menghitung jumlah maksimum koneksi yang HAProxy memungkinkan backend. Misalnya, Anda memiliki server

aplikasi Rails pada instance kecil dengan `backend` nilai 4, yang berarti AWS OpsWorks Stacks akan mengonfigurasi empat proses Rails untuk instance itu. Jika Anda menggunakan `maxcon_factor_rails_app` nilai default 7, HAProxy akan menangani 28 (4* 7) koneksi ke server Rails.

maxcon_factor_nodejs_app

Faktor maxcon untuk server aplikasi Node.js (angka). Nilai default-nya adalah `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app]
```

maxcon_factor_nodejs_app_ssl

Faktor maxcon untuk server aplikasi Node.js dengan SSL (angka). Nilai default-nya adalah `10`.

```
node[:haproxy][:maxcon_factor_nodejs_app_ssl]
```

maxcon_factor_php_app

Faktor maxcon untuk server aplikasi PHP (angka). Nilai default-nya adalah `10`.

```
node[:haproxy][:maxcon_factor_php_app]
```

maxcon_factor_php_app_ssl

Faktor maxcon untuk server aplikasi PHP dengan SSL (angka). Nilai default-nya adalah `10`.

```
node[:haproxy][:maxcon_factor_php_app_ssl]
```

maxcon_factor_rails_app

Faktor maxcon untuk server aplikasi Rails (angka). Nilai default-nya adalah 7.

```
node[:haproxy][:maxcon_factor_rails_app]
```

maxcon_factor_rails_app_ssl

Faktor maxcon untuk server aplikasi Rails dengan SSL (angka). Nilai default-nya adalah 7.

```
node[:haproxy][:maxcon_factor_rails_app_ssl]
```

maxcon_factor_static

Faktor maxcon untuk server web statis (angka). Nilai default-nya adalah 15.

```
node[:haproxy][:maxcon_factor_static]
```

maxcon_factor_static_ssl

Faktor maxcon untuk server web statis dengan SSL (angka). Nilai default-nya adalah 15.

```
node[:haproxy][:maxcon_factor_static_ssl]
```

Atribut memcached

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Atribut ini hanya tersedia di tumpukan Linux.

`memcached`Atribut menentukan konfigurasi server Memcached. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

memori	max_connections	pid_file
port	start_command	stop_perintah
user		

memori

Memori maksimum yang digunakan, dalam MB (angka). Nilai default-nya adalah `512`.

```
node[:memcached][:memory]
```

max_connections

Jumlah maksimum koneksi (string). Nilai default-nya adalah `'4096'`.

```
node[:memcached][:max_connections]
```

pid_file

File yang berisi ID proses daemon (string). Nilai bawaannya adalah `'var/run/memcached.pid'`.

```
node[:memcached][:pid_file]
```

port

Port untuk mendengarkan (nomor). Nilai default-nya adalah `11211`.

```
node[:memcached][:port]
```

start_command

Perintah mulai (string). Nilai default-nya adalah `'/etc/init.d/memcached start'`.

```
node[:memcached][:start_command]
```

stop_perintah

Perintah stop (string). Nilai default-nya adalah `'/etc/init.d/memcached stop'`.

```
node[:memcached][:stop_command]
```

user

Pengguna (string). Nilai default-nya adalah `'nobody'`.

```
node[:memcached][:user]
```

Atribut mysql

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Atribut ini hanya tersedia di tumpukan Linux.

`mysql`Atribut menentukan konfigurasi master MySQL. Untuk informasi selengkapnya, lihat Variabel Sistem Server. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

Basedir	bind_address	klien
conf_dir	confd_dir	datadir
grants_path	mysql_bin	mysqladmin_bin
pid_file	port	root_group
server_root_password	soket	Atribut yang dapat disetel

Basedir

Direktori dasar (string). Nilai default-nya adalah `'/usr'`.

```
node[:mysql][:basedir]
```

bind_address

Alamat yang MySQL mendengarkan pada (string). Nilai default-nya adalah `'0.0.0.0'`.

```
node[:mysql][:bind_address]
```

klien

Daftar klien (daftar string).

```
node[:mysql][:clients]
```

conf_dir

Direktori yang berisi file konfigurasi (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'/etc'`
- Ubuntu: `'/etc/mysql'`

```
node[:mysql][:conf_dir]
```

confd_dir

Direktori yang berisi file konfigurasi tambahan (string). Nilai default-nya adalah `'/etc/mysql/conf.d'`.

```
node[:mysql][:confd_dir]
```

datadir

Direktori data (string). Nilai default-nya adalah `'/var/lib/mysql'`.

```
node[:mysql][:datadir]
```

grants_path

Lokasi tabel hibah (string). Nilai default-nya adalah `'/etc/mysql_grants.sql'`.

```
node[:mysql][:grants_path]
```

mysql_bin

Lokasi binari mysql (string). Nilai default-nya adalah `'/usr/bin/mysql'`.

```
node[:mysql][:mysql_bin]
```

mysqladmin_bin

Lokasi mysqladmin (string). Nilai default-nya adalah `'/usr/bin/mysqladmin'`.

```
node[:mysql][:mysqladmin_bin]
```

pid_file

File yang berisi ID proses daemon (string). Nilai bawaannya adalah `'/var/run/mysqld/mysqld.pid'`.

```
node[:mysql][:pid_file]
```

port

Port tempat server mendengarkan (nomor). Nilai default-nya adalah `3306`.

```
node[:mysql][:port]
```

root_group

Grup root (string). Nilai default-nya adalah `'root'`.

```
node[:mysql][:root_group]
```

server_root_password

Kata sandi root server (string). Nilai default dihasilkan secara acak.

```
node[:mysql][:server_root_password]
```

soket

Lokasi file soket (string). Nilai default-nya adalah `'/var/lib/mysql/mysql.sock'`. Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'/var/lib/mysql/mysql.sock'`
- Ubuntu: `'/var/run/mysqld/mysqld.sock'`

```
node[:mysql][:socket]
```

Atribut yang dapat disetel

Atribut yang dapat disetel digunakan untuk penyetelan kinerja.

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

Jumlah maksimum permintaan yang belum dibayar (string). Nilai default-nya adalah `'128'`.

```
node[:mysql][:tunable][:back_log]
```

innodb_additional_mem_pool_size

Ukuran pool yang digunakan Innodb untuk menyimpan struktur data internal (string). Nilai default-nya adalah `'20M'`.

```
node[:mysql][:tunable][:innodb_additional_mem_pool_size]
```

innodb_buffer_pool_size

Ukuran kolam buffer Innodb (string). Nilai atribut diatur oleh AWS OpsWorks Stacks dan bergantung pada jenis instance, tetapi Anda dapat menggantinya dengan menggunakan JSON kustom atau file atribut kustom.

```
node[:mysql][:tunable][:innodb_buffer_pool_size]
```

innodb_flush_log_at_trx_commit

Seberapa sering Innodb menyiram buffer log (string). Nilai default-nya adalah `'2'`. Untuk informasi selengkapnya, lihat innodb_flush_log_at_trx_commit.

```
node[:mysql][:tunable][:innodb_flush_log_at_trx_commit]
```

innodb_lock_wait_timeout

Jumlah maksimum waktu, dalam hitungan detik, bahwa transaksi Innodb menunggu kunci baris (string). Nilai default-nya adalah `'50'`.

```
node[:mysql][:tunable][:innodb_lock_wait_timeout]
```

key_buffer

Ukuran buffer indeks (string). Nilai default-nya adalah `'250M'`.

```
node[:mysql][:tunable][:key_buffer]
```

log_slow_queries

Lokasi file log slow-query (string). Nilai default-nya adalah `'/var/log/mysql/mysql-slow.log'`.

```
node[:mysql][:tunable][:log_slow_queries]
```

long_query_time

Waktu, dalam hitungan detik, diperlukan untuk menunjuk kueri sebagai kueri panjang (string). Nilai default-nya adalah `'1'`.

```
node[:mysql][:tunable][:long_query_time]
```

max_allowed_packet

Ukuran paket maksimum yang diizinkan (string). Nilai default-nya adalah `'32M'`.

```
node[:mysql][:tunable][:max_allowed_packet]
```

max_connections

Jumlah maksimum koneksi klien bersamaan (string). Nilai default-nya adalah `'2048'`.

```
node[:mysql][:tunable][:max_connections]
```

max_heap_table_size

Ukuran maksimum `MEMORY` tabel yang dibuat pengguna (string). Nilai default-nya adalah `'32M'`.

```
node[:mysql][:tunable][:max_heap_table_size]
```

net_read_timeout

Jumlah waktu, dalam hitungan detik, untuk menunggu lebih banyak data dari koneksi (string). Nilai default-nya adalah `'30'`.

```
node[:mysql][:tunable][:net_read_timeout]
```

net_write_timeout

Jumlah waktu, dalam hitungan detik, untuk menunggu blok ditulis ke koneksi (string). Nilai default-nya adalah `'30'`.

```
node[:mysql][:tunable][:net_write_timeout]
```

query_cache_limit

Ukuran maksimum kueri cache individu (string). Nilai default-nya adalah `'2M'`.

```
node[:mysql][:tunable][:query_cache_limit]
```

query_cache_size

Ukuran cache query (string). Nilai default-nya adalah `'128M'`.

```
node[:mysql][:tunable][:query_cache_size]
```

query_cache_type

Jenis cache query (string). Kemungkinan nilainya adalah sebagai berikut:

- `'0'`: Tidak ada caching atau pengambilan data cache.
- `'1'`: Pernyataan cache yang tidak dimulai dengan`SELECT SQL_NO_CACHE`.
- `'2'`: Pernyataan cache yang dimulai dengan`SELECT SQL_CACHE`.

Nilai default-nya adalah `'1'`.

```
node[:mysql][:tunable][:query_cache_type]
```

thread_cache_size

Jumlah thread klien yang di-cache untuk digunakan kembali (string). Nilai default-nya adalah `'8'`.

```
node[:mysql][:tunable][:thread_cache_size]
```

thread_stack

Ukuran tumpukan untuk setiap utas (string). Nilai default-nya adalah `'192K'`.

```
node[:mysql][:tunable][:thread_stack]
```

wait_timeout

Jumlah waktu, dalam hitungan detik, untuk menunggu koneksi noninteraktif. Nilai defaultnya adalah `'180'` (string).

```
node[:mysql][:tunable][:wait_timeout]
```

table_cache

Jumlah tabel terbuka (string). Nilai default-nya adalah `'2048'`.

```
node[:mysql][:tunable][:table_cache]
```

nginx Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Atribut ini hanya tersedia di tumpukan Linux.

`nginx`Atribut menentukan konfigurasi Nginx. Untuk informasi selengkapnya, lihat Indeks Petunjuk. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

biner	dir	gzip
gzip_comp_level	gzip_disable	gzip_http_version
gzip_proxied	gzip_static	gzip_types
gzip_vary	keepalive	keepalive_timeout
log_dir	user	server_names_hash_bucket_size
worker_processes	worker_connections	

biner

Lokasi binari Nginx (string). Nilai default-nya adalah `'/usr/sbin/nginx'`.

```
node[:nginx][:binary]
```

dir

Lokasi file seperti file konfigurasi (string). Nilai default-nya adalah `'/etc/nginx'`.

```
node[:nginx][:dir]
```

gzip

Apakah kompresi gzip diaktifkan (string). Nilai yang mungkin adalah `'on'` dan `'off'`. Nilai default-nya adalah `'on'`.

> **Warning**
>
> Kompresi dapat menimbulkan risiko keamanan. Untuk sepenuhnya menonaktifkan kompresi, atur atribut ini sebagai berikut:
>
> ```
> node[:nginx][:gzip] = 'off'
> ```

```
node[:nginx][:gzip]
```

gzip_comp_level

Tingkat kompresi, yang dapat berkisar dari 1-9, dengan 1 sesuai dengan kompresi terkecil (string). Nilai default-nya adalah `'2'`.

```
node[:nginx][:gzip_comp_level]
```

gzip_disable

Menonaktifkan kompresi gzip untuk agen pengguna tertentu (string). Nilainya adalah ekspresi reguler dan nilai defaultnya adalah`'MSIE [1-6].(?!.*SV1)'`.

```
node[:nginx][:gzip_disable]
```

gzip_http_version

Mengaktifkan kompresi gzip untuk versi HTTP tertentu (string). Nilai default-nya adalah `'1.0'`.

```
node[:nginx][:gzip_http_version]
```

gzip_proxied

Apakah dan bagaimana mengompres respons terhadap permintaan proxy, yang dapat mengambil salah satu nilai berikut (string):

- `'off'`: jangan kompres permintaan proksi
- `'expired'`: kompres jika header Kedaluwarsa mencegah caching
- `'no-cache'`: kompres jika header Cache-Control disetel ke “no-cache”
- `'no-store'`: kompres jika header Cache-Control disetel ke “no-store”
- `'private'`: kompres jika header Cache-Control disetel ke “pribadi”
- `'no_last_modified'`: kompres jika Last-Modified tidak disetel
- `'no_etag'`: kompres jika permintaan tidak memiliki header ETag
- `'auth'`: kompres jika permintaan menyertakan header Otorisasi
- `'any'`: kompres semua permintaan yang diproksi

Nilai default-nya adalah `'any'`.

```
node[:nginx][:gzip_proxied]
```

gzip_static

Apakah modul statis gzip diaktifkan (string). Nilai yang mungkin adalah `'on'` dan `'off'`. Nilai default-nya adalah `'on'`.

```
node[:nginx][:gzip_static]
```

gzip_types

Daftar tipe MIME yang akan dikompresi (daftar string). Nilai default-nya adalah `['text/plain', 'text/html', 'text/css', 'application/x-javascript', 'text/xml', 'application/xml', 'application/xml+rss', 'text/javascript']`.

```
node[:nginx][:gzip_types]
```

gzip_vary

Apakah akan mengaktifkan header `Vary:Accept-Encoding` respons (string). Nilai yang mungkin adalah `'on'` dan `'off'`. Nilai default-nya adalah `'on'`.

```
node[:nginx][:gzip_vary]
```

keepalive

Apakah akan mengaktifkan koneksi keep-alive (string). Nilai yang mungkin adalah `'on'` dan `'off'`. Nilai default-nya adalah `'on'`.

```
node[:nginx][:keepalive]
```

keepalive_timeout

Jumlah waktu maksimum, dalam hitungan detik, koneksi tetap hidup tetap terbuka (angka). Nilai default-nya adalah 65.

```
node[:nginx][:keepalive_timeout]
```

log_dir

Lokasi file log (string). Nilai default-nya adalah `'/var/log/nginx'`.

```
node[:nginx][:log_dir]
```

user

Pengguna (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `'www-data'`
- Ubuntu: `'nginx'`

```
node[:nginx][:user]
```

server_names_hash_bucket_size

Ukuran bucket untuk tabel hash dari nama server, yang dapat diatur ke32,64, atau 128 (nomor). Nilai default-nya adalah 64.

```
node[:nginx][:server_names_hash_bucket_size]
```

worker_processes

Jumlah proses pekerja (jumlah). Nilai default-nya adalah `10`.

```
node[:nginx][:worker_processes]
```

worker_connections

Jumlah maksimum koneksi pekerja (angka). Nilai default-nya adalah `1024`. Jumlah maksimum klien diatur ke`worker_processes * worker_connections`.

```
node[:nginx][:worker_connections]
```

opsworks_berkshelf Atribut

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Atribut ini hanya tersedia di tumpukan Linux.

`opsworks_berkshelf`Atribut menentukan konfigurasi Berkshelf. Untuk informasi lebih lanjut, lihat Berkshelf. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

debug

Apakah akan menyertakan informasi debugging Berkshelf di log Chef (Boolean). Nilai default-nya adalah `false`.

```
node['opsworks_berkshelf][['debug']
```

opsworks_java Atribut

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Atribut ini hanya tersedia di tumpukan Linux.

`opsworks_java`Atribut menentukan konfigurasi server Tomcat. Untuk informasi selengkapnya, lihat Referensi Konfigurasi Apache Tomcat. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

sumber data	java_app_server_version	java_shared_lib_dir
jvm_pkg Atribut	custom_pkg_location_url_debian	java_home_basedir
custom_pkg_location_url_rhel	gunakan_custom_pkg_location	jvm_options
jvm_version	Atribut tomcat	

sumber data

Satu set atribut yang mendefinisikan nama sumber daya JNDI (string). Untuk informasi selengkapnya tentang cara menggunakan atribut ini, lihatMenyebarkan Aplikasi JSP dengan Back-End Database. Nilai default adalah hash kosong, yang dapat diisi dengan pemetaan khusus

antara nama pendek aplikasi dan nama JNDI. Untuk informasi selengkapnya, lihat Menyebarkan Aplikasi JSP dengan Back-End Database.

```
node['opsworks_java']['datasources']
```

java_app_server_version

Versi server aplikasi Java (nomor). Nilai default-nya adalah 7. Anda dapat mengganti atribut ini untuk menentukan versi 6. Jika Anda menginstal JDK nondefault, atribut ini diabaikan.

```
node['opsworks_java']['java_app_server_version']
```

java_shared_lib_dir

Direktori untuk perpustakaan bersama Java (string). Nilai default-nya adalah `/usr/share/java`.

```
node['opsworks_java']['java_shared_lib_dir']
```

jvm_pkg Atribut

Satu set atribut yang dapat Anda ganti untuk menginstal JDK nondefault.

gunakan_custom_pkg_location

Apakah akan menginstal JDK kustom alih-alih OpenJDK (Boolean). Nilai default-nya adalah `false`.

```
node['opsworks_java']['jvm_pkg']['use_custom_pkg_location']
```

custom_pkg_location_url_debian

Lokasi paket JDK yang akan diinstal pada instance Ubuntu (string). Nilai defaultnya adalah `'http://aws.amazon.com/'`, yang hanya merupakan nilai inisialisasi tanpa arti yang tepat. Jika Anda ingin menginstal JDK nondefault, Anda harus mengganti atribut ini dan mengaturnya ke URL yang sesuai.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_debian']
```

custom_pkg_location_url_rhel

Lokasi paket JDK yang akan diinstal di Amazon Linux dan instance RHEL (string). Nilai defaultnya adalah `'http://aws.amazon.com/'`, yang hanya merupakan nilai inisialisasi

tanpa arti yang tepat. Jika Anda ingin menginstal JDK nondefault, Anda harus mengganti atribut ini dan mengaturnya ke URL yang sesuai.

```
node['opsworks_java']['jvm_pkg']['custom_pkg_location_url_rhel']
```

java_home_basedir

Direktori yang paket JDK akan diekstraksi ke (string). Nilai default-nya adalah `/usr/local`. Anda tidak perlu menentukan pengaturan ini untuk paket RPM; mereka menyertakan struktur direktori lengkap.

```
node['opsworks_java']['jvm_pkg']['java_home_basedir']
```

jvm_options

Opsi baris perintah JVM, yang memungkinkan Anda menentukan pengaturan seperti ukuran tumpukan (string). Satu set opsi yang umum adalah`-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC`. Nilai default adalah tidak ada opsi.

```
node['opsworks_java']['jvm_options']
```

jvm_version

Versi OpenJDK (nomor). Nilai default-nya adalah 7. Anda dapat mengganti atribut ini untuk menentukan OpenJDK versi 6. Jika Anda menginstal JDK nondefault, atribut ini diabaikan.

```
node['opsworks_java']['jvm_version']
```

Atribut tomcat

Satu set atribut yang dapat Anda timpa untuk menginstal konfigurasi Tomcat default.

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_pengkodean	gunakan_ssl_connector	gunakan_threadpool
userdatabase_pathname		

ajp_port

Port AJP (nomor). Nilai default-nya adalah `8009`.

```
node['opsworks_java']['tomcat]['ajp_port']
```

apache_tomcat_bind_mod

Modul proxy (string). Nilai default-nya adalah `proxy_http`. Anda dapat mengganti atribut ini untuk menentukan modul proxy AJP,. `proxy_ajp`

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_mod']
```

apache_tomcat_bind_path

Jalur pengikat Apache-Tomcat (string). Nilai default-nya adalah `/`. Anda tidak boleh mengganti atribut ini; mengubah jalur bind dapat menyebabkan aplikasi berhenti bekerja.

```
node['opsworks_java']['tomcat]['apache_tomcat_bind_path']
```

auto_deploy

Apakah akan autodeploy (Boolean). Nilai default-nya adalah `true`.

```
node['opsworks_java']['tomcat]['auto_deploy']
```

connection_timeout

Batas waktu koneksi, dalam milidetik (angka). Nilai defaultnya adalah `20000` (20 detik).

```
node['opsworks_java']['tomcat]['connection_timeout']
```

mysql_connector_jar

File JAR pustaka konektor MySQL (string). Nilai bawaannya adalah `mysql-connector-java.jar`.

```
node['opsworks_java']['tomcat]['mysql_connector_jar']
```

port

Port standar (nomor). Nilai default-nya adalah `8080`.

```
node['opsworks_java']['tomcat]['port']
```

secure_port

Port aman (nomor). Nilai default-nya adalah `8443`.

```
node['opsworks_java']['tomcat]['secure_port']
```

shutdown_port

Port shutdown (nomor). Nilai default-nya adalah `8005`.

```
node['opsworks_java']['tomcat]['shutdown_port']
```

threadpool_max_threads

Jumlah maksimum utas di kumpulan utas (angka). Nilai default-nya adalah `150`.

```
node['opsworks_java']['tomcat]['threadpool_max_threads']
```

threadpool_min_spare_threads

Jumlah minimum utas cadangan di kumpulan utas (nomor). Nilai default-nya adalah `4`.

```
node['opsworks_java']['tomcat]['threadpool_min_spare_threads']
```

unpack_wars

Apakah akan membongkar file WAR (Boolean). Nilai default-nya adalah `true`.

```
node['opsworks_java']['tomcat]['unpack_wars']
```

uri_pengkodean

Pengkodean URI (string). Nilai default-nya adalah `UTF-8`.

```
node['opsworks_java']['tomcat]['uri_encoding']
```

gunakan_ssl_connector

Apakah akan menggunakan konektor SSL (Boolean). Nilai default-nya adalah `false`.

```
node['opsworks_java']['tomcat]['use_ssl_connector']
```

gunakan_threadpool

Apakah akan menggunakan thread pool (Boolean). Nilai default-nya adalah `false`.

```
node['opsworks_java']['tomcat]['use_threadpool']
```

userdatabase_pathname

Nama path database pengguna (string). Nilai default-nya adalah `conf/tomcat-users.xml`.

```
node['opsworks_java']['tomcat]['userdatabase_pathname']
```

Atribut passenger_apache2

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Atribut ini hanya tersedia di tumpukan Linux.

`passenger_apache2`Atribut menentukan konfigurasi Penumpang Phusion. Untuk informasi selengkapnya, lihat panduan pengguna Penumpang Phusion, versi Apache. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

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	versi	

friendly_error_pages

Apakah akan menampilkan halaman kesalahan ramah jika aplikasi gagal memulai (string). Atribut ini dapat diatur ke 'on' atau 'off'; nilai default adalah 'off'.

```
node[:passenger][:friendly_error_pages]
```

gem_bin

Lokasi binari Permata (string). Nilai default-nya adalah `'/usr/local/bin/gem'`.

```
node[:passenger][:gem_bin]
```

gems_path

Jalur permata (string). Nilai default tergantung pada versi Ruby. Misalnya:

- Ruby versi 1.8: `'/usr/local/lib/ruby/gems/1.8/gems'`
- Ruby versi 1.9: `'/usr/local/lib/ruby/gems/1.9.1/gems'`

```
node[:passenger][:gems_path]
```

high_performance_mode

Apakah akan menggunakan mode performa tinggi (string) Penumpang. Nilai yang mungkin adalah `'on'` dan `'off'`. Nilai default-nya adalah `'off'`.

```
node[:passenger][:high_performance_mode ]
```

root_path

Direktori root Passenger (string). Nilai default tergantung pada versi Ruby dan Passenger. Dalam sintaks Chef, nilainya adalah`"#{node[:passenger][:gems_path]}/passenger-#{passenger[:version]}"`.

```
node[:passenger][:root_path]
```

max_instances_per_app

Jumlah maksimum proses aplikasi per aplikasi (nomor). Nilai default-nya adalah `0`. Untuk informasi selengkapnya, lihat PassengerMaxInstancesPerApp.

```
node[:passenger][:max_instances_per_app]
```

max_pool_size

Jumlah maksimum prosesor aplikasi (angka). Nilai default-nya adalah 8. Untuk informasi selengkapnya, lihat PassengerMaxPoolSize.

```
node[:passenger][:max_pool_size]
```

max_requests

Jumlah maksimum permintaan (nomor). Nilai default-nya adalah `0`.

```
node[:passenger][:max_requests]
```

module_path

Jalur modul (string). Nilai defaultnya adalah sebagai berikut:

- Amazon Linux dan RHEL: `"#{node['apache']['libexecdir']}/mod_passenger.so"`

- Ubuntu: `"#{passenger[:root_path]}/ext/apache2/mod_passenger.so"`

```
node[:passenger][:module_path]
```

pool_idle_time

Waktu maksimum, dalam hitungan detik, proses aplikasi bisa idle (angka). Nilai defaultnya adalah `14400` (4 jam). Untuk informasi selengkapnya, lihat PassengerPoolIdleTime.

```
node[:passenger][:pool_idle_time]
```

rails_app_spawner_idle_time

Waktu idle maksimum untuk spawner aplikasi Rails (angka). Jika atribut ini disetel ke nol, spawner aplikasi tidak akan habis waktu. Nilai default-nya adalah `0`. Untuk informasi lebih lanjut, lihat Metode Pemijahan Dijelaskan.

```
node[:passenger][:rails_app_spawner_idle_time]
```

rails_framework_spawner_idle_time

Waktu idle maksimum untuk spawner kerangka kerja Rails (angka). Jika atribut ini disetel ke nol, spawner kerangka kerja tidak habis waktu. Nilai default-nya adalah `0`. Untuk informasi lebih lanjut, lihat Metode Pemijahan Dijelaskan.

```
node[:passenger][:rails_framework_spawner_idle_time]
```

rails_spawn_method

Metode spawn Rails (string). Nilai default-nya adalah `'smart-lv2'`. Untuk informasi lebih lanjut, lihat Metode Pemijahan Dijelaskan.

```
node[:passenger][:rails_spawn_method]
```

ruby_bin

Lokasi binari Ruby (string). Nilai default-nya adalah `'/usr/local/bin/ruby'`.

```
node[:passenger][:ruby_bin]
```

ruby_wrapper_bin

Lokasi skrip pembungkus Ruby (string). Nilai default-nya adalah `'/usr/local/bin/ruby_gc_wrapper.sh'`.

```
node[:passenger][:ruby_wrapper_bin]
```

stat_throttle_rate

Tarif di mana Penumpang melakukan pemeriksaan sistem file (nomor). Nilai defaultnya adalah5, yang berarti bahwa pemeriksaan akan dilakukan paling banyak sekali setiap 5 detik. Untuk informasi selengkapnya, lihat PassengerStatThrottleRate .

```
node[:passenger][:stat_throttle_rate]
```

versi

Versi (string). Nilai default-nya adalah `'3.0.9'`.

```
node[:passenger][:version]
```

Atribut ruby

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Atribut ini hanya tersedia di tumpukan Linux.

`rubyAtribut` menentukan versi Ruby yang digunakan oleh aplikasi. Perhatikan bahwa penggunaan atribut berubah dengan diperkenalkannya versi semantik di Ruby 2.1. Untuk informasi selengkapnya

tentang cara menentukan versi, termasuk contoh, lihatVersi Ruby. Untuk detail lengkap tentang cara AWS OpsWorks Stacks menentukan versi Ruby, lihat file atribut bawaan, ruby.rb. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

full_version

Nomor versi lengkap (string). Anda tidak harus mengganti atribut ini. Sebagai gantinya, gunakan [:opsworks] [:ruby_version] dan atribut versi patch yang sesuai untuk menentukan versi.

```
[:ruby][:full_version]
```

major_version

Nomor versi utama (string). Anda tidak harus mengganti atribut ini. Sebagai gantinya, gunakan [:opsworks] [:ruby_version] untuk menentukan versi utama.

```
[:ruby][:major_version]
```

minor_version

Nomor versi minor (string). Anda tidak harus mengganti atribut ini. Sebagai gantinya, gunakan [:opsworks] [:ruby_version] untuk menentukan versi minor.

```
[:ruby][:minor_version]
```

tambalan

Level patch (string). Atribut ini berlaku untuk Ruby versi 2.0.0 dan sebelumnya. Untuk versi Ruby yang lebih baru, gunakan `patch_version` atribut.

```
[:ruby][:patch]
```

Nomor tambalan harus diawali olehp. Misalnya, Anda akan menggunakan JSON kustom berikut untuk menentukan tingkat patch 484.

```
{
  "ruby":{"patch":"p484"}
}
```

patch_version

Nomor patch (string). Atribut ini berlaku untuk Ruby versi 2.1 dan yang lebih baru. Untuk versi Ruby sebelumnya, gunakan `patch` atribut.

```
[:ruby][:patch_version]
```

pkgrelease

Nomor rilis paket (string).

```
[:ruby][:pkgrelease]
```

Atribut unicorn

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Atribut ini hanya tersedia di tumpukan Linux.

`unicorn`Atribut menentukan konfigurasi Unicorn. Untuk informasi selengkapnya, lihat Unicorn: :Configurator. Untuk informasi selengkapnya tentang cara mengganti atribut bawaan untuk menentukan nilai kustom, lihatMengesampingkan Atribut.

accept_filter	tumpukan	delay
tcp_nodelay	tcp_nopush	preload_app
batas waktu	berusaha	versi

worker_processes		

accept_filter

Filter terima, `'httpready'` atau `'dataready'` (string). Nilai default-nya adalah `'httpready'`.

```
node[:unicorn][:accept_filter]
```

tumpukan

Jumlah maksimum permintaan yang dapat disimpan antrian (nomor). Nilai default-nya adalah `1024`.

```
node[:unicorn][:backlog]
```

delay

Jumlah waktu, dalam hitungan detik, menunggu untuk mencoba lagi mengikat soket (angka). Nilai default-nya adalah `0.5`.

```
node[:unicorn][:delay]
```

preload_app

Apakah akan melakukan pramuat aplikasi sebelum melakukan fork proses pekerja (Boolean). Nilai default-nya adalah `true`.

```
node[:unicorn][:preload_app]
```

tcp_nodelay

Apakah akan menonaktifkan algoritma Nagle untuk soket TCP (Boolean). Nilai default-nya adalah `true`.

```
node[:unicorn][:tcp_nodelay]
```

tcp_nopush

Apakah akan mengaktifkan TCP_CORK (Boolean). Nilai default-nya adalah `false`.

```
node[:unicorn][:tcp_nopush]
```

batas waktu

Jumlah waktu maksimum, dalam hitungan detik, yang diizinkan untuk digunakan oleh seorang pekerja untuk setiap permintaan (nomor). Pekerja yang melebihi nilai batas waktu dihentikan. Nilai default-nya adalah `60`.

```
node[:unicorn][:timeout]
```

berusaha

Jumlah maksimum kali untuk mencoba lagi mengikat ke soket (angka). Nilai default-nya adalah `5`.

```
node[:unicorn][:tries]
```

versi

Versi Unicorn (string). Nilai default-nya adalah `'4.7.0'`.

```
node[:unicorn][:version]
```

worker_processes

Jumlah proses pekerja (jumlah). Nilai defaultnya adalah`max_pool_size`, jika ada, dan `4` sebaliknya.

```
node[:unicorn][:worker_processes]
```

Pemecahan Masalah Chef 11.10 dan Versi Sebelumnya untuk Linux

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Untuk informasi pemecahan masalah tambahan, lihat. Panduan Debugging dan Pemecahan Masalah

Log Chef untuk Chef 11.10 dan Versi Sebelumnya untuk Linux

AWS OpsWorks Stacks menyimpan log Chef setiap instance di `/var/lib/aws/opsworks/chef` direktorinya. Anda memerlukan hak istimewa sudo untuk mengakses direktori ini. Log untuk setiap proses ada dalam file bernama*YYYY*-*MM*-*DD*-*HH*-*MM*-*SS*-*NN*.log.

Untuk informasi selengkapnya, lihat informasi berikut:

- Melihat Log Chef dengan Konsol
- Melihat Log Chef dengan CLI atau API
- Menafsirkan Log Chef
- Kesalahan Log Chef Umum

Menggunakan AWS OpsWorks Stacks dengan Layanan AWS Lainnya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat memiliki server aplikasi yang berjalan di tumpukan AWS OpsWorks Stacks menggunakan berbagai layanan AWS yang tidak terintegrasi langsung dengan AWS OpsWorks Stacks. Misalnya, Anda dapat meminta server aplikasi Anda menggunakan Amazon RDS sebagai database back-end. Anda dapat mengakses layanan tersebut dengan menggunakan pola umum berikut:

1. Buat dan konfigurasikan layanan AWS dengan menggunakan konsol AWS, API, atau CLI dan catat data konfigurasi yang diperlukan aplikasi untuk mengakses layanan, seperti nama host atau port.
2. Buat satu atau lebih resep khusus untuk mengonfigurasi aplikasi sehingga dapat mengakses layanan.

   Resep memperoleh data konfigurasi dari konfigurasi tumpukan dan atribut JSON penerapan yang Anda tentukan dengan JSON kustom sebelum menjalankan resep.
3. Tetapkan resep kustom ke peristiwa siklus hidup Deploy pada lapisan server aplikasi.
4. Buat objek JSON kustom yang menetapkan nilai yang sesuai ke atribut data konfigurasi dan menambahkannya ke konfigurasi tumpukan dan penyebaran JSON Anda.
5. Menyebarkan aplikasi ke tumpukan.

   Deployment menjalankan resep kustom, yang menggunakan nilai data konfigurasi yang Anda tentukan di JSON kustom untuk mengonfigurasi aplikasi sehingga dapat mengakses layanan.

Bagian ini menjelaskan cara agar server aplikasi AWS OpsWorks Stacks mengakses berbagai layanan AWS. Ini mengasumsikan bahwa Anda sudah terbiasa dengan buku masak Chef dan bagaimana resep dapat menggunakan atribut tumpukan dan konfigurasi JSON untuk mengonfigurasi aplikasi, biasanya dengan membuat file konfigurasi. Jika tidak, Anda harus terlebih dahulu membaca Buku Masak dan Resep danMenyesuaikan Tumpukan AWS OpsWorks.

Topik

- Menggunakan Back-end Data Store
- Menggunakan ElastiCache Redis sebagai Toko Nilai Kunci-Dalam Memori
- Menggunakan Amazon S3 Bucket
- Menggunakan AWS CodePipeline dengan AWS OpsWorks Stacks

Menggunakan Back-end Data Store

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan server aplikasi biasanya mencakup server database untuk menyediakan penyimpanan data back-end. AWS OpsWorks Stacks menyediakan dukungan terintegrasi untuk server MySQL melalui lapisan MySQL dan untuk beberapa jenis server database melalui lapisan Amazon Relational Database Service (Amazon RDS). Namun, Anda dapat dengan mudah menyesuaikan tumpukan agar server aplikasi menggunakan server database lain seperti Amazon DynamoDB atau MongoDB. Topik ini menjelaskan prosedur dasar untuk menghubungkan server aplikasi ke server database AWS. Ini menggunakan tumpukan dan aplikasi dari Memulai dengan Chef 11 Linux Stacks untuk menunjukkan bagaimana menghubungkan server aplikasi PHP secara manual ke database RDS. Meskipun contoh didasarkan pada tumpukan Linux, prinsip-prinsip dasar juga berlaku untuk tumpukan Windows. Untuk contoh cara menggabungkan server database MongoDB ke dalam tumpukan, lihat Menyebarkan MongoDB dengan. OpsWorks

Note

Topik ini menggunakan Amazon RDS sebagai contoh yang nyaman. Namun, jika Anda ingin menggunakan database Amazon RDS dengan tumpukan Anda, jauh lebih mudah untuk menggunakan lapisan Amazon RDS.

Topik

- Cara Mengatur Koneksi Database
- Cara Menghubungkan Instans Server Aplikasi ke Amazon RDS

Cara Mengatur Koneksi Database

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda mengatur koneksi antara server aplikasi dan basis data back-endnya dengan menggunakan resep khusus. Resep mengkonfigurasi server aplikasi sesuai kebutuhan, biasanya dengan membuat file konfigurasi. Resep mendapatkan data koneksi seperti nama host dan database dari sekumpulan atribut dalam konfigurasi tumpukan dan atribut penerapan yang diinstal AWS OpsWorks Stacks pada setiap instance.

Misalnya, Langkah 2 Memulai dengan Chef 11 Linux Stacks didasarkan pada tumpukan bernama MyStack dengan dua lapisan, PHP App Server dan MySQL, masing-masing dengan satu instance. Anda menerapkan aplikasi bernama Simple PHPApp ke instance PHP App Server yang menggunakan database pada instance MySQL sebagai penyimpanan data back-end. Saat Anda menerapkan aplikasi, AWS OpsWorks Stacks menginstal konfigurasi tumpukan dan atribut penerapan yang berisi informasi koneksi database. Contoh berikut menunjukkan atribut koneksi database, direpresentasikan sebagai JSON:

```
{
  ...
  "deploy": {
    "simplephpapp": {
      ...
      "database": {
        "reconnect": true,
        "password": null,
        "username": "root",
        "host": null,
        "database": "simplephpapp"
        ...
      },
      ...
    }
  }
}
```

Nilai atribut disediakan oleh AWS OpsWorks Stacks, dan dihasilkan atau berdasarkan informasi yang diberikan pengguna.

Untuk memungkinkan Simple PHPApp mengakses penyimpanan data, Anda harus mengatur koneksi antara server aplikasi PHP dan database MySQL dengan menetapkan resep kustom `appsetup.rb` bernama ke peristiwa siklus hidup Deploy lapisan PHP App Server. Saat Anda menerapkan SimplePHPApp, AWS OpsWorks Stacks berjalan`appsetup.rb`, yang membuat file

konfigurasi bernama `db-connect.php` yang mengatur koneksi, seperti yang ditunjukkan dalam kutipan berikut.

```
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
```

Variabel yang mencirikan koneksi—`host`,`user`, dan sebagainya—ditetapkan nilai yang sesuai dari atribut JSON deploy. `[:deploy][:app_name][:database]` Untuk mempermudah, contoh mengasumsikan bahwa Anda telah membuat tabel bernama`aurler`, sehingga nama tabel diwakili oleh `[:phpapp][:dbtable]` dalam file atribut buku masak.

Resep ini sebenarnya dapat menghubungkan server aplikasi PHP ke server database MySQL manapun, bukan hanya anggota lapisan MySQL. Untuk menggunakan server MySQL yang berbeda, Anda hanya perlu mengatur atribut ke nilai `[:database]` yang sesuai untuk server Anda, yang dapat Anda lakukan dengan menggunakan JSON kustom. AWS OpsWorks Tumpukan kemudian menggabungkan atribut dan nilai tersebut ke dalam konfigurasi tumpukan dan atribut penerapan dan `appsetup.rb` menggunakannya untuk membuat template yang mengatur koneksi. Untuk informasi selengkapnya tentang penggantian konfigurasi tumpukan dan penerapan JSON, lihat. Mengesampingkan Atribut

Cara Menghubungkan Instans Server Aplikasi ke Amazon RDS

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan cara menyesuaikan MyStack dari Memulai dengan Chef 11 Linux Stacks agar server aplikasi PHP terhubung ke instance RDS.

Topik

- Buat Database MySQL Amazon RDS
- Sesuaikan Stack untuk Connect ke Database RDS

Buat Database MySQL Amazon RDS

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sekarang Anda siap untuk membuat database RDS untuk contoh menggunakan Amazon RDS Console Launch DB Instance Wizard. Prosedur berikut adalah ringkasan singkat dari detail penting. Untuk penjelasan rinci tentang cara membuat database, lihat Memulai Amazon RDS.

Untuk membuat database Amazon RDS

1. Jika ini adalah pertama kalinya Anda membuat database RDS, klik Mulai Sekarang. Jika tidak, klik Dasbor RDS di panel navigasi, lalu klik Luncurkan Instans DB.

2. Pilih MySQL Community Edition sebagai instance DB.
3. Untuk Apakah Anda berencana menggunakan database ini untuk tujuan produksi? pilih Tidak, contoh ini... , yang cukup untuk contoh. Untuk penggunaan produksi, Anda mungkin ingin memilih Ya, gunakan Penyebaran Multi-AZ... . Klik Langkah Selanjutnya.
4. Pada halaman Tentukan Detail DB, tentukan pengaturan berikut:

   - Kelas Instans DB: db.t2.micro
   - Penerapan Multi-AZ: Tidak
   - Penyimpanan yang Dialokasikan: GB **5**
   - Pengidentifikasi Instans DB: **rdsexample**
   - Nama Pengguna Master: **opsworksuser**
   - Master Password: Tentukan kata sandi yang sesuai dan rekam untuk digunakan nanti.

   Terima pengaturan default untuk opsi lain dan klik Langkah Berikutnya.
5. Pada halaman Konfigurasi Pengaturan Lanjut, tentukan pengaturan berikut:

   - Di bagian Jaringan & Keamanan, untuk Grup Keamanan VPC, pilih phpsecgroup (VPC)
   - Di bagian Opsi Database, untuk Nama Database, ketik **rdsexampledb**
   - Di bagian Backup, atur Periode Retensi Cadangan ke 0 untuk keperluan panduan ini.

   Terima pengaturan default untuk opsi lain dan klik Luncurkan Instans DB.
6. Pilih Lihat Instans DB Anda untuk melihat daftar instans DB.
7. Pilih contoh rdsexample dalam daftar dan klik panah untuk mengungkapkan titik akhir instance dan detail lainnya. Catat titik akhir untuk digunakan nanti. Ini akan menjadi sesuatu seperti`rdsexample.c6c8mntzhgv0.us-west-2.rds.amazonaws.com:3306`. Cukup catat nama DNS; Anda tidak memerlukan nomor port.
8. Gunakan alat seperti MySQL Workbench untuk membuat tabel `urler` bernama dalam database dengan menggunakan perintah SQL `rdsexampledb` berikut:

```
CREATE TABLE urler(id INT UNSIGNED NOT NULL AUTO_INCREMENT,author VARCHAR(63) NOT
 NULL,message TEXT,PRIMARY KEY (id))
```

Sesuaikan Stack untuk Connect ke Database RDS

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Setelah Anda membuat instance RDS untuk digunakan sebagai database back-end untuk server aplikasi PHP, Anda dapat menyesuaikan dari. MyStack Memulai dengan Chef 11 Linux Stacks

Untuk menghubungkan PHP App Server ke database RDS

1. Buka konsol AWS OpsWorks Stacks dan buat tumpukan dengan lapisan PHP App Server yang berisi satu instance dan gunakan SimplePHPApp, seperti yang dijelaskan dalam. Memulai dengan Chef 11 Linux Stacks Tumpukan ini menggunakan version1 dari SimplePHPApp, yang tidak menggunakan koneksi database.
2. Perbarui konfigurasi tumpukan untuk menggunakan buku masak khusus yang menyertakan `appsetup.rb` resep, serta file template dan atribut terkait.
    1. Atur Gunakan buku masak Chef khusus ke Ya.
    2. Atur jenis Repositori ke Git dan URL Repositori ke. `git://github.com/amazonwebservices/opsworks-example-cookbooks.git`
3. Tambahkan yang berikut ini ke kotak JSON Custom Chef tumpukan untuk menetapkan data koneksi RDS ke `[:database]` atribut yang `appsetup.rb` digunakan untuk membuat file konfigurasi.

```
{
  "deploy": {
    "simplephpapp": {
      "database": {
        "username": "opsworksuser",
        "password": "your_password",
        "database": "rdsexampledb",
        "host": "rds_endpoint",
```

```
        "adapter": "mysql"
      }
    }
  }
}
```

Gunakan nilai atribut berikut:

- username: Nama pengguna master yang Anda tentukan saat Anda membuat instance RDS.

  Contoh ini menggunakan `opsworksuser`.
- password: Kata sandi utama yang Anda tentukan saat Anda membuat instance RDS.

  Isi kata sandi yang Anda tentukan.
- database: Database yang Anda buat ketika Anda membuat instance RDS.

  Contoh ini menggunakan `rdsexampledb`.
- host: Titik akhir instans RDS, yang Anda dapatkan dari konsol RDS saat Anda membuat instance di bagian sebelumnya. Jangan sertakan nomor port.
- adaptor: Adaptor.

  Contoh RDS untuk contoh ini menggunakan MySQL, sehingga adaptor diatur ke. `mysql` Berbeda dengan atribut lainnya, adaptor tidak digunakan oleh`appsetup.rb`. Ini malah digunakan oleh resep Konfigurasi bawaan lapisan PHP App Server untuk membuat file konfigurasi yang berbeda.

4. Edit PHPApp konfigurasi Simple untuk menentukan versi Simple PHPApp yang menggunakan database back-end, sebagai berikut:

   - Root dokumen: Setel opsi ini ke`web`.
   - Cabang/Revisi: Setel opsi ini ke. `version2`

   Biarkan opsi yang tersisa tidak berubah.
5. Edit lapisan PHP App Server untuk mengatur koneksi database dengan `phpapp::appsetup` menambahkan resep Deploy layer.
6. Menyebarkan PHPApp versi Sederhana yang baru.

7. Ketika Simple PHPApp digunakan, jalankan aplikasi dengan membuka halaman Instances dan mengklik alamat IP publik php-app1 instans. Anda akan melihat halaman berikut di browser Anda, yang memungkinkan Anda memasukkan teks dan menyimpannya di database.



> **Note**
>
> Jika tumpukan Anda memiliki lapisan MySQL AWS OpsWorks , Stacks secara otomatis menetapkan data koneksi yang sesuai ke atribut. `[:database]` Namun, jika Anda menetapkan JSON kustom ke tumpukan yang mendefinisikan nilai yang berbeda, mereka akan mengganti `[:database]` nilai default. Karena `[:deploy]` atribut diinstal pada setiap contoh, resep apa pun yang bergantung pada `[:database]` atribut akan menggunakan data koneksi khusus, bukan data lapisan MySQL untuk. Jika Anda ingin lapisan server aplikasi tertentu menggunakan data koneksi kustom, tetapkan JSON kustom ke acara Deploy layer, dan batasi penyebaran itu ke lapisan itu. Untuk informasi selengkapnya tentang cara menggunakan atribut penerapan, lihatMenerapkan Aplikasi. Untuk informasi selengkapnya tentang mengganti atribut bawaan AWS OpsWorks Stacks, lihat. Mengesampingkan Atribut

Menggunakan ElastiCache Redis sebagai Toko Nilai Kunci-Dalam Memori

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Topik ini didasarkan pada tumpukan Linux, tetapi tumpukan Windows juga dapat menggunakan Amazon ElastiCache (ElastiCache). Untuk contoh bagaimana cara menggunakan ElastiCache instance Windows, lihat ElastiCache sebagai ASP.NET Session Store.

Anda sering dapat meningkatkan kinerja server aplikasi dengan menggunakan server caching untuk menyediakan penyimpanan nilai kunci dalam memori untuk item kecil data seperti string. Amazon ElastiCache adalah layanan AWS yang memudahkan untuk menyediakan dukungan caching untuk server aplikasi Anda, menggunakan mesin caching Memcached atau Redis. AWS OpsWorks Stacks menyediakan dukungan bawaan untuk Memcached. Namun, jika Redis lebih sesuai dengan kebutuhan Anda, Anda dapat menyesuaikan tumpukan Anda sehingga server aplikasi Anda menggunakan ElastiCache Redis.

Topik ini memandu Anda melalui proses dasar penyediaan dukungan caching ElastiCache Redis untuk tumpukan Linux, menggunakan server aplikasi Rails sebagai contoh. Ini mengasumsikan bahwa Anda sudah memiliki aplikasi Ruby on Rails yang sesuai. Untuk informasi lebih lanjut tentang ElastiCache, lihat Apa itu Amazon ElastiCache? .

Topik

- Langkah 1: Buat ElastiCache Redis Cluster
- Langkah 2: Siapkan Rails Stack
- Langkah 3: Buat dan Terapkan Buku Masak Kustom
- Langkah 4: Tetapkan Resep ke Acara LifeCycle

- Langkah 5: Tambahkan Informasi Akses ke Konfigurasi Stack JSON
- Langkah 6: Terapkan dan jalankan Aplikasi

Langkah 1: Buat ElastiCache Redis Cluster

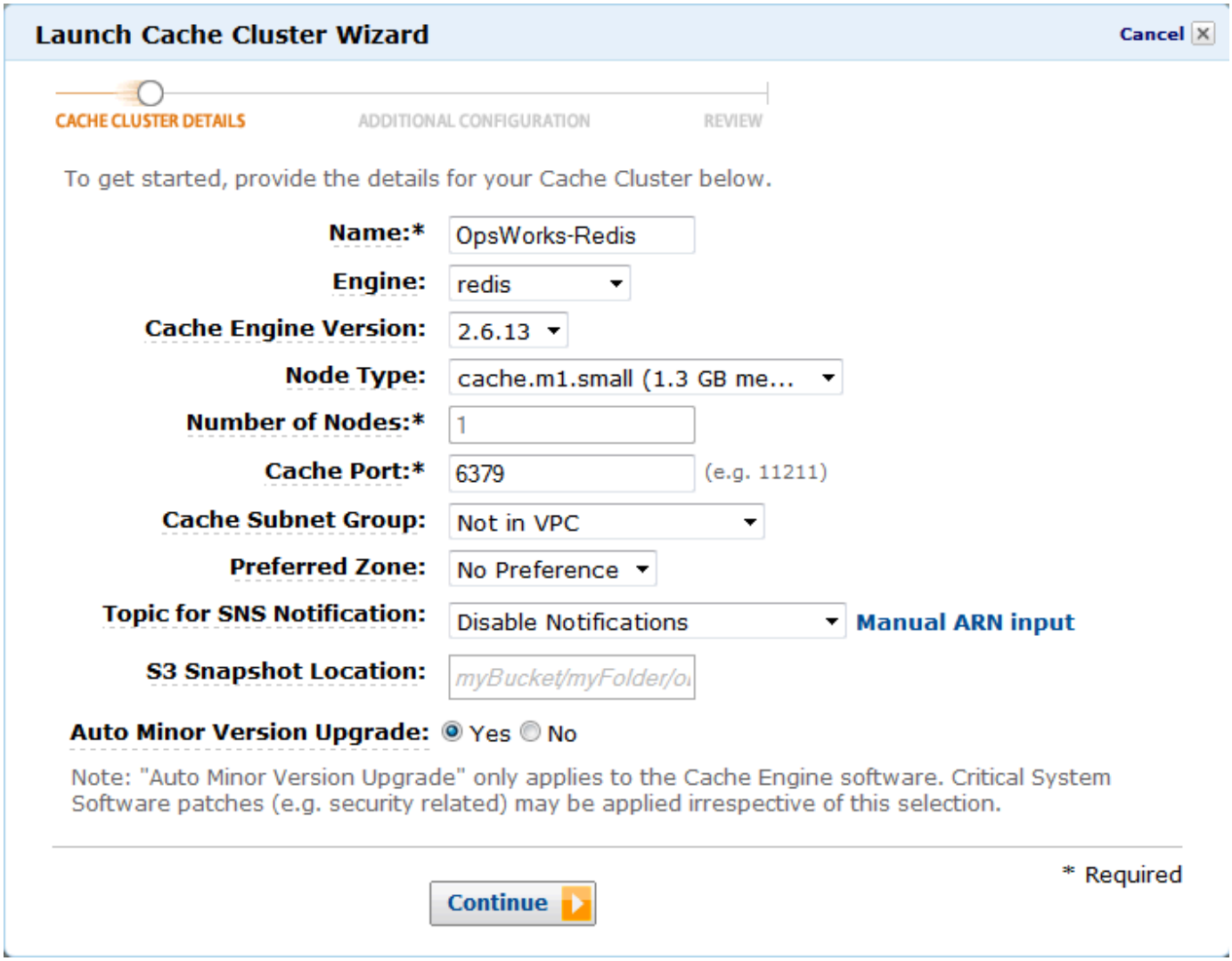
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda harus terlebih dahulu membuat klaster Amazon ElastiCache Redis dengan menggunakan ElastiCache konsol, API, atau CLI. Berikut ini menjelaskan cara menggunakan konsol untuk membuat cluster.

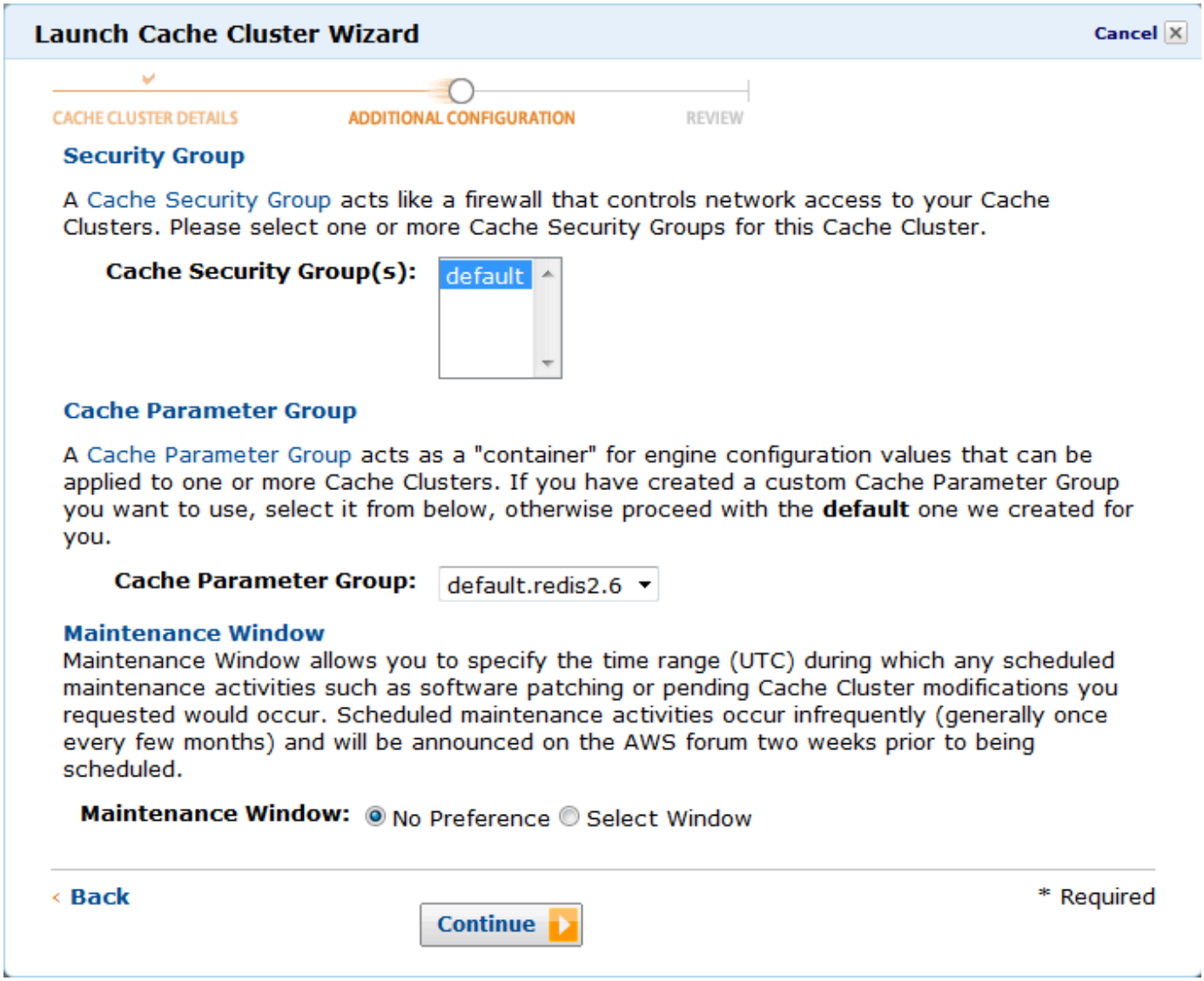
Untuk membuat cluster ElastiCache Redis

1. Buka ElastiCachekonsol dan klik Luncurkan Cache Cluster untuk memulai wizard Cache Cluster.
2. Pada halaman Cache Cluster Details, lakukan hal berikut:
   - Tetapkan Nama ke nama server cache Anda.

     Contoh ini menggunakan OpsWorks -Redis.
   - Atur Engine ke Redis.
   - Setel Topik untuk Pemberitahuan SNS untuk Menonaktifkan Pemberitahuan.
   - Terima default untuk pengaturan lain dan klik Lanjutkan.



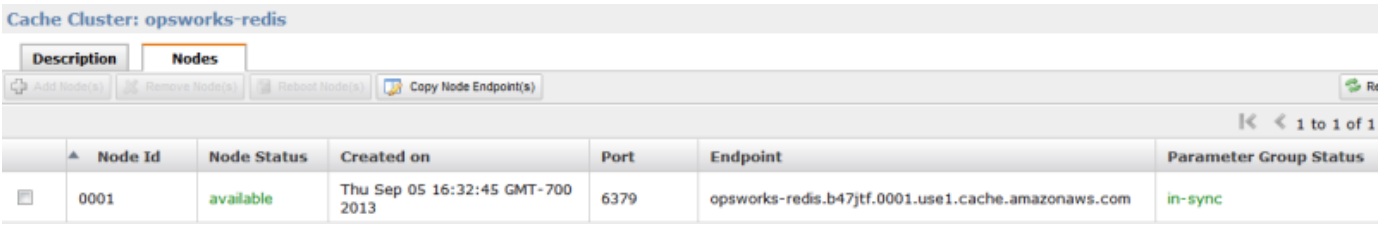
3. Pada halaman Konfigurasi Tambahan, terima default dan klik Lanjutkan.



4. Klik Luncurkan Cache Cluster untuk membuat cluster.

> **Important**
>
> Grup keamanan cache default cukup untuk contoh ini, tetapi untuk penggunaan produksi Anda harus membuat satu yang sesuai untuk lingkungan Anda. Untuk informasi selengkapnya, lihat Mengelola Grup Keamanan Cache.

5. Setelah cluster dimulai, klik nama untuk membuka halaman detail dan klik tab Nodes. Catat nilai Port dan Endpoint cluster untuk digunakan nanti.



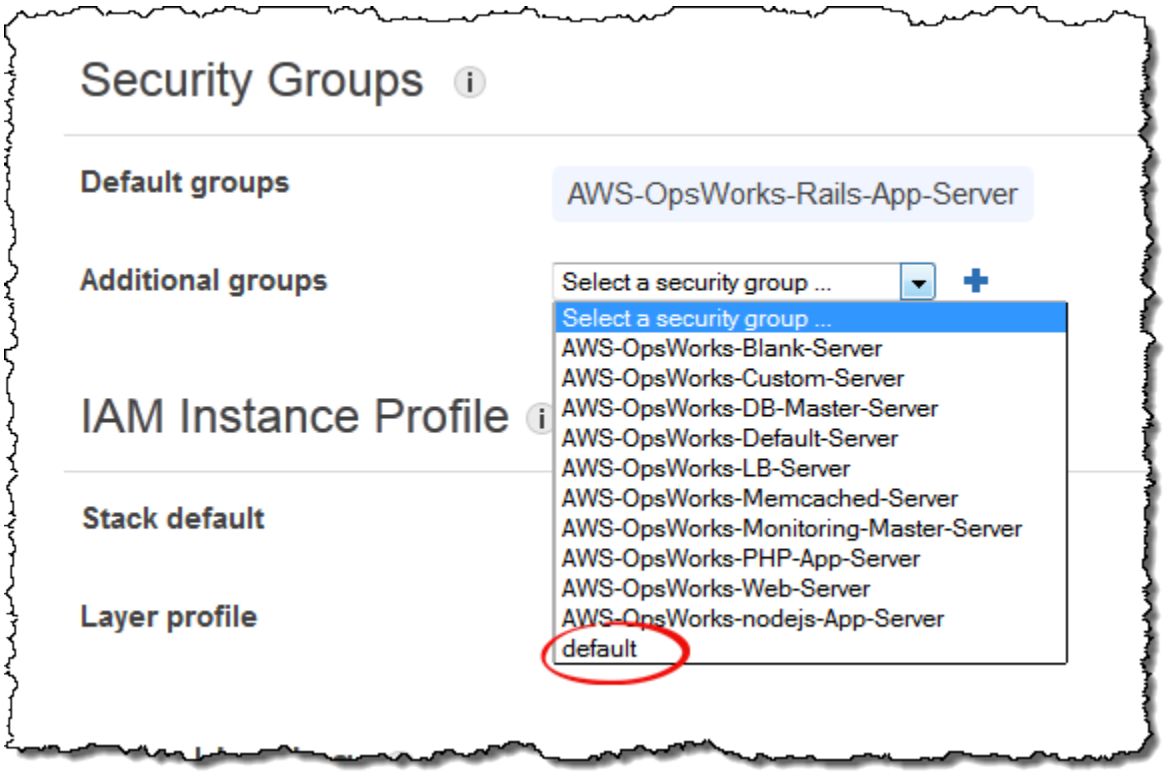
Langkah 2: Siapkan Rails Stack

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Selain membuat tumpukan yang mendukung lapisan Rails App Server, Anda juga harus mengkonfigurasi grup keamanan lapisan sehingga server Rails dapat berkomunikasi dengan baik dengan server Redis.

Untuk mengatur tumpukan

1. Buat tumpukan baru—dinamai **`RedisStack`** untuk contoh ini—dan tambahkan layer Rails App Server. Anda dapat menggunakan pengaturan default untuk keduanya. Untuk informasi selengkapnya, silakan lihat Membuat Tumpukan Baru dan Membuat OpsWorks Layer .
2. Pada halaman Layers, untuk Rails App Server, klik Keamanan dan kemudian klik Edit.
3. Buka bagian Grup Keamanan dan tambahkan grup keamanan ElastiCache klaster ke Grup tambahan. Untuk contoh ini, pilih grup keamanan default, klik + untuk menambahkannya ke layer, dan klik Simpan untuk menyimpan konfigurasi baru.



4. Tambahkan instance ke layer Rails App Server dan mulai. Untuk informasi selengkapnya tentang cara menambahkan dan memulai instance, lihatMenambahkan Instance ke Layer.

Langkah 3: Buat dan Terapkan Buku Masak Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Saat ini, tumpukan belum cukup fungsional; Anda perlu mengaktifkan aplikasi Anda untuk mengakses server Redis. Pendekatan yang paling fleksibel adalah menempatkan file YAMM dengan informasi akses di `config` subfolder aplikasi. Aplikasi kemudian dapat memperoleh informasi dari file. Dengan menggunakan pendekatan ini, Anda dapat mengubah informasi koneksi tanpa menulis ulang dan memindahkan aplikasi. Untuk contoh ini, file harus diberi nama `redis.yml` dan berisi nama host ElastiCache cluster dan port, sebagai berikut:

```
host: cache-cluster-hostname
port: cache-cluster-port
```

Anda dapat menyalin file ini secara manual ke server Anda, tetapi pendekatan yang lebih baik adalah menerapkan resep Chef untuk menghasilkan file, dan meminta AWS OpsWorks Stacks menjalankan resep di setiap server. Resep koki adalah aplikasi Ruby khusus yang digunakan AWS OpsWorks Stacks untuk melakukan tugas pada instance seperti menginstal paket atau membuat file konfigurasi. Resep dikemas dalam buku masak, yang dapat berisi beberapa resep dan file terkait seperti templat untuk file konfigurasi. Buku masak ditempatkan di repositori, seperti GitHub, dan harus memiliki struktur direktori standar. Jika Anda belum memiliki repositori buku masak khusus, lihat Repositori untuk informasi tentang cara mengaturnya.

Untuk contoh ini, tambahkan buku masak bernama `redis-config` ke repositori buku masak Anda dengan konten berikut:

```
my_cookbook_repository
  redis-config
    recipes
      generate.rb
    templates
      default
        redis.yml.erb
```

`recipes`Folder berisi resep bernama`generate.rb`, yang menghasilkan file konfigurasi aplikasi dari`redis.yml.erb`, sebagai berikut:

```
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
```

Resepnya bergantung pada data dari konfigurasi tumpukan AWS OpsWorks Stacks dan objek JSON penerapan, yang diinstal pada setiap instance dan berisi informasi terperinci tentang tumpukan dan aplikasi apa pun yang diterapkan. `deploy`Node objek memiliki struktur sebagai berikut:

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Node deploy berisi satu set objek JSON yang disematkan, satu untuk setiap aplikasi yang di-deploy, yang diberi nama dengan nama pendek aplikasi. Setiap objek aplikasi berisi sekumpulan atribut yang menentukan konfigurasi aplikasi, seperti root dokumen dan jenis aplikasi. Untuk daftar atribut penerapan, lihatmenyebarkan Atribut. Resep dapat menggunakan sintaks atribut Chef untuk mewakili konfigurasi tumpukan dan penerapan nilai JSON. Misalnya, `[:deploy][:app1][:application]` mewakili nama pendek aplikasi app1.

Untuk setiap aplikasi di`[:deploy]`, resep mengeksekusi blok kode terkait, yang `deploy_config` mewakili atribut app. Resep pertama disetel `app_root` ke direktori root aplikasi,`[:deploy][:app_name][:deploy_to]/current`. Kemudian menggunakan sumber daya template Chef untuk menghasilkan file konfigurasi dari `redis.yml.erb` dan menempatkannya di file`app_root/config`.

File konfigurasi biasanya dibuat dari templat, dengan banyak jika tidak sebagian besar pengaturan ditentukan oleh atribut Chef. Dengan atribut Anda dapat mengubah pengaturan menggunakan JSON kustom, seperti yang dijelaskan nanti, alih-alih menulis ulang file template.Template berisi yang berikut`redis.yml.erb`:

```
host: <%= @redis[:host] %>
port: <%= @redis[:port] || 6379 %>
```

Elemen **<%... %>** adalah placeholder yang mewakili nilai atribut.

- `<%= @redis[:host] %>`mewakili nilai`redis[:host]`, yang merupakan nama host cluster cache.
- `<%= @redis[:port] || 6379 %>`mewakili nilai `redis[:port]` atau, jika atribut itu tidak didefinisikan, nilai port default, 6379.

Sumber `template` daya berfungsi sebagai berikut:

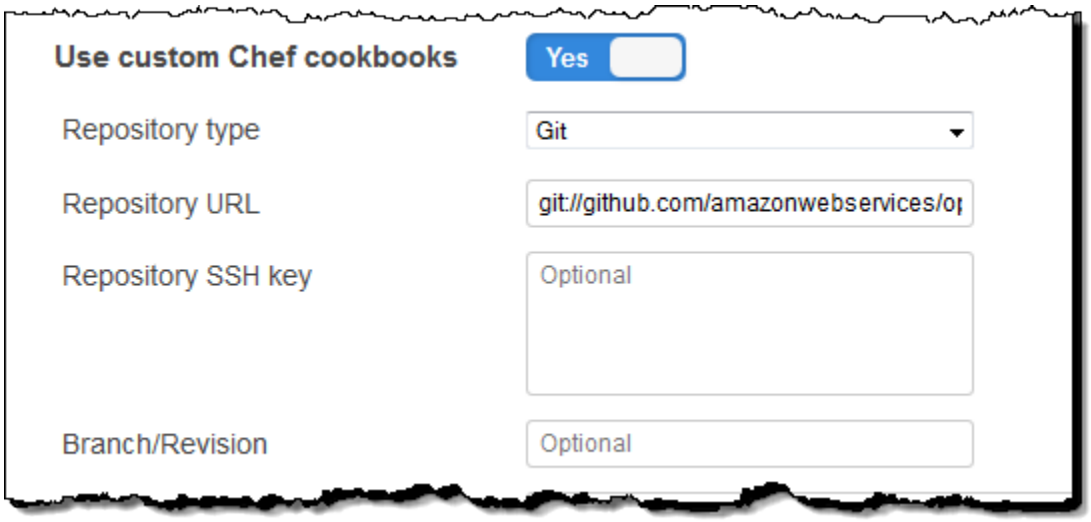
- `source`dan `cookbook` tentukan nama template dan buku masak, masing-masing.
- `mode`,`group`, dan `owner` berikan file konfigurasi hak akses yang sama dengan aplikasi.
- `variables`Bagian ini menetapkan `@redis` variabel yang digunakan dalam template, ke nilai `[:redis]` atribut aplikasi.

  Nilai `[:redis]` atribut ditetapkan dengan menggunakan JSON kustom, seperti yang dijelaskan nanti; ini bukan salah satu atribut aplikasi standar.
- `not_if`Arahan memastikan bahwa resep tidak menghasilkan file konfigurasi jika sudah ada.

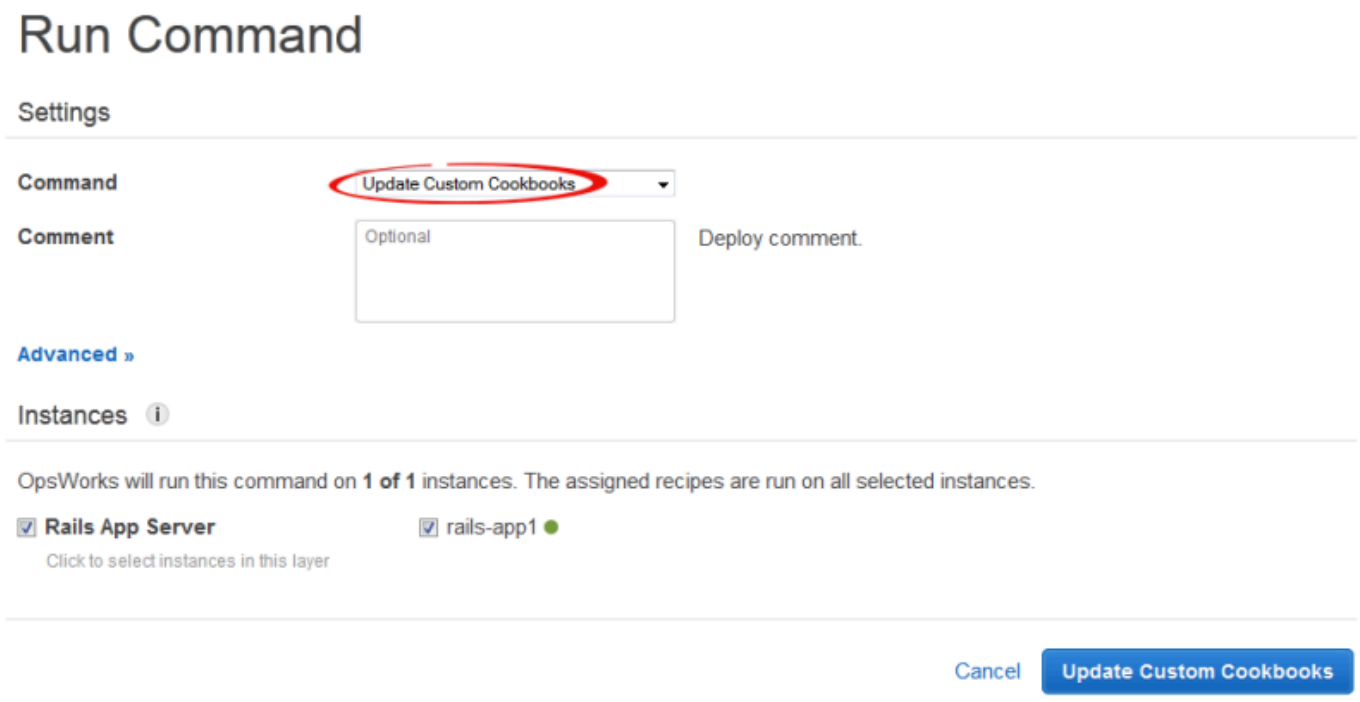
Setelah Anda menulis buku masak, Anda harus menerapkannya ke cache buku masak masing-masing instance. Operasi ini tidak menjalankan resep; itu hanya menginstal buku masak baru pada instance tumpukan. Anda biasanya menjalankan resep dengan menetapkannya ke peristiwa siklus hidup lapisan, seperti yang dijelaskan nanti.

Untuk menyebarkan buku masak kustom Anda

1. Pada halaman AWS OpsWorks Stacks Stack, klik Stack Settings dan kemudian Edit.
2. Di bagian Manajemen Konfigurasi, atur Gunakan buku masak Chef kustom ke Ya, masukkan informasi repositori buku masak, dan klik Simpan untuk memperbarui konfigurasi tumpukan.



3. Pada halaman Stack, klik Run Command, pilih perintah Update Custom Cookbooks stack, dan klik Update Custom Cookbooks untuk menginstal buku masak baru di cache buku masak instance.



Jika Anda memodifikasi buku masak Anda, jalankan Perbarui Buku Masak Kustom lagi untuk menginstal versi yang diperbarui. Untuk informasi lebih lanjut tentang prosedur ini, lihatMemasang Buku Masak Kustom.

Langkah 4: Tetapkan Resep ke Acara LifeCycle

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

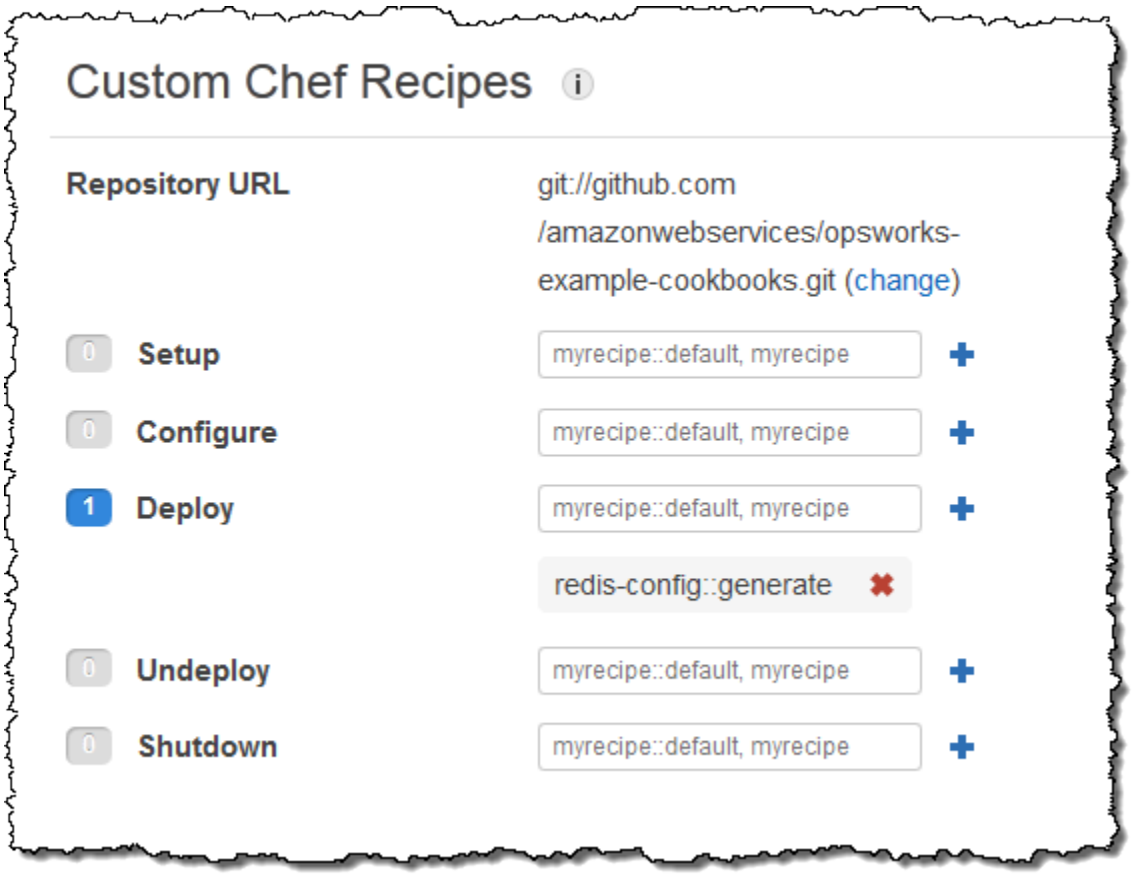
Anda dapat menjalankan resep khusus secara manual, tetapi pendekatan terbaik biasanya adalah memiliki AWS OpsWorks Stacks menjalankannya secara otomatis. Setiap lapisan memiliki satu set resep bawaan yang ditetapkan masing-masing dari lima peristiwa siklus hidup —Setup, Configure, Deploy, Undeploy, dan Shutdown. Setiap kali peristiwa terjadi untuk sebuah instance, AWS OpsWorks Stacks menjalankan resep terkait untuk setiap lapisan instance, yang menangani tugas yang sesuai. Misalnya, saat instance selesai booting, AWS OpsWorks Stacks memicu peristiwa Penyiapan. Acara ini menjalankan resep Setup layer terkait, yang biasanya menangani tugas-tugas seperti menginstal dan mengkonfigurasi paket.

Anda dapat meminta AWS OpsWorks Stacks menjalankan resep khusus pada instance layer dengan menetapkan resep ke peristiwa siklus hidup yang sesuai. Untuk contoh ini, Anda harus menetapkan `generate.rb` resep ke event Deploy layer Rails App Server. AWS OpsWorks Stacks kemudian akan menjalankannya pada instance layer selama startup, setelah resep Setup selesai, dan setiap kali Anda menerapkan aplikasi. Untuk informasi selengkapnya, lihat Resep Menjalankan Secara Otomatis.

Untuk menetapkan resep ke event Deploy layer Rails App Server

1. Pada halaman AWS OpsWorks Stacks Layers, untuk Rails App Server, klik Resep dan kemudian klik Edit..
2. Di bawah Resep Koki Kustom, tambahkan nama resep yang sepenuhnya memenuhi syarat ke acara penerapan dan klik +. Nama resep yang sepenuhnya memenuhi syarat menggunakan *`cookbookname`*`::`*`recipename`* format, di mana *`recipename`* tidak termasuk

`.rb` ekstensi. Untuk contoh ini, nama yang sepenuhnya memenuhi syarat adalah`redis-config::generate`. Kemudian klik Simpan untuk memperbarui konfigurasi lapisan.



Langkah 5: Tambahkan Informasi Akses ke Konfigurasi Stack JSON

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

`generate.rb`Resepnya tergantung pada sepasang konfigurasi tumpukan dan penerapan atribut JSON yang mewakili nama host dan port server Redis. Meskipun atribut ini adalah bagian dari `[:deploy]` namespace standar, mereka tidak secara otomatis ditentukan oleh AWS OpsWorks

Stacks. Sebagai gantinya, Anda mendefinisikan atribut dan nilainya dengan menambahkan objek JSON khusus ke tumpukan. Contoh berikut menunjukkan JSON kustom untuk contoh ini.

Untuk menambahkan informasi akses ke konfigurasi tumpukan dan penerapan JSON

1. Pada halaman AWS OpsWorks Stacks Stack, klik Stack Settings dan kemudian Edit.
2. Di bagian Manajemen Konfigurasi, tambahkan informasi akses ke kotak Custom Chef JSON. Seharusnya terlihat seperti contoh berikut, dengan modifikasi ini:
   - Ganti `elasticache_redis_example` dengan nama pendek aplikasi Anda.
   - Ganti `port` nilai `host` dan dengan nilai untuk instance server ElastiCache Redis yang Anda buat. Langkah 1: Buat ElastiCache Redis Cluster

```
{
  "deploy": {
     "elasticache_redis_example": {
       "redis": {
         "host": "mycluster.XXXXXXXXX.amazonaws.com",
         "port": "6379"
       }
     }
  }
}
```



Keuntungan dari pendekatan ini adalah Anda dapat mengubah port atau nilai host kapan saja tanpa menyentuh buku masak khusus Anda. AWS OpsWorks Stacks menggabungkan JSON kustom ke dalam JSON bawaan dan menginstalnya pada instance stack untuk semua peristiwa siklus hidup berikutnya. Aplikasi kemudian dapat mengakses nilai atribut dengan menggunakan sintaks simpul Chef, seperti yang dijelaskan dalamLangkah 3: Buat dan Terapkan Buku Masak Kustom. Lain kali Anda menerapkan aplikasi, AWS OpsWorks Stacks akan menginstal konfigurasi tumpukan dan penerapan JSON yang berisi definisi baru, dan `generate.rb` akan membuat file konfigurasi dengan nilai host dan port yang diperbarui.

> **Note**
>
> `[:deploy]`secara otomatis menyertakan atribut untuk setiap aplikasi yang diterapkan, jadi sudah `[:deploy][elasticache_redis_example]` ada di tumpukan dan konfigurasi JSON. Namun, `[:deploy][elasticache_redis_example]` tidak menyertakan `[:redis]` atribut, mendefinisikannya dengan JSON kustom mengarahkan AWS OpsWorks Stacks untuk menambahkan atribut tersebut. `[:deploy][elasticache_redis_example]` Anda juga dapat menggunakan JSON khusus untuk mengganti atribut yang ada. Untuk informasi selengkapnya, lihat Mengesampingkan Atribut.

Langkah 6: Terapkan dan jalankan Aplikasi

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Contoh ini mengasumsikan bahwa Anda memiliki aplikasi Ruby on Rails yang menggunakan Redis. Untuk mengakses file konfigurasi, Anda dapat menambahkan `redis` permata ke Gemfile Anda dan membuat penginisialisasi Rails sebagai berikut: `config/initializers/redis.rb`

```
REDIS_CONFIG = YAML::load_file(Rails.root.join('config', 'redis.yml'))
$redis = Redis.new(:host => REDIS_CONFIG['host'], :port => REDIS_CONFIG['port'])
```

Kemudian buat aplikasi untuk mewakili aplikasi Anda dan terapkan ke instance layer Rails App Server, yang memperbarui kode aplikasi dan berjalan `generate.rb` untuk menghasilkan file konfigurasi. Ketika Anda menjalankan aplikasi, itu akan menggunakan instance ElastiCache Redis sebagai penyimpanan nilai kunci dalam memori.

Menggunakan Amazon S3 Bucket

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Aplikasi sering menggunakan Amazon Simple Storage Service (Amazon Storage Service) untuk menyimpan item besar seperti gambar atau file media lainnya. Meskipun AWS OpsWorks Stacks tidak menyediakan dukungan terintegrasi untuk Amazon S3, Anda dapat dengan mudah menyesuaikan tumpukan untuk memungkinkan aplikasi Anda menggunakan penyimpanan Amazon S3. Topik ini memandu Anda melalui proses dasar menyediakan akses Amazon S3 ke aplikasi, menggunakan tumpukan Linux dengan server aplikasi PHP sebagai contoh. Prinsip dasar juga berlaku untuk tumpukan Windows.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Cara Menghapus S3 Bucket? .

Topik

- Langkah 1: Buat Bucket Amazon S3
- Langkah 2: Buat Tumpukan Server Aplikasi PHP
- Langkah 3: Buat dan Men-deploy Buku Masak Kustom
- Langkah 4: Tetapkan Resep ke Acara LifeCycle
- Langkah 5: Tambahkan Informasi Akses ke Konfigurasi Stack dan Atribut Deployment
- Langkah 6: Men-deploy dan Jalankan PhotoApp

Langkah 1: Buat Bucket Amazon S3

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Pertama-tama Anda harus membuat bucket Amazon S3. Anda dapat melakukan ini secara langsung dengan menggunakan konsol Amazon S3, API, atau CLI, tetapi cara yang lebih sederhana untuk membuat sumber daya sering menggunakan template. AWS CloudFormation Template berikut membuat bucket Amazon S3 untuk contoh ini dan menyiapkan profil instance dengan peran IAM yang memberikan akses tak terbatas ke bucket. Anda kemudian dapat menggunakan pengaturan lapisan untuk melampirkan profil instance ke instance server aplikasi stack, yang memungkinkan aplikasi mengakses bucket, seperti yang dijelaskan nanti. Kegunaan profil instans tidak terbatas pada Amazon S3; mereka berharga untuk mengintegrasikan berbagai layanan 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" }
       }
   }
}
```

Beberapa hal terjadi ketika Anda meluncurkan template:

- Sumber `AWS::S3::Bucket` daya membuat bucket Amazon S3.
- `AWS::IAM::InstanceProfile`Sumber daya membuat profil instance yang akan ditugaskan ke instance server aplikasi.
- `AWS::IAM::Role`Sumber daya menciptakan peran profil instance.

- Sumber `AWS::IAM::Policy` daya menetapkan izin peran untuk mengizinkan akses tidak terbatas ke bucket Amazon S3.
- `Outputs`Bagian ini menampilkan nama profil bucket dan instance di AWS CloudFormation konsol setelah Anda meluncurkan template.

  Anda akan memerlukan nilai-nilai ini untuk mengatur tumpukan dan aplikasi Anda.

Untuk informasi selengkapnya tentang cara membuat AWS CloudFormation templat, lihat Pelajari Dasar-Dasar Template.

Untuk membuat bucket Amazon S3

1. Salin templat contoh ke file teks pada sistem Anda.

   Contoh ini mengasumsikan bahwa file tersebut diberi nama`appserver.template`.
2. Buka AWS CloudFormationkonsol dan pilih Create Stack.
3. Di kotak Stack Name, masukkan nama tumpukan.

   Contoh ini mengasumsikan bahwa namanya adalah**`AppServer`**.
4. Pilih Unggah file templat, pilih Jelajahi, pilih `appserver.template` file yang Anda buat di Langkah 1, lalu pilih Langkah Berikutnya.
5. Pada halaman Tentukan Parameter, pilih Saya mengakui bahwa templat ini dapat membuat sumber daya IAM, lalu pilih Langkah Berikutnya pada setiap halaman wizard hingga Anda mencapai akhir. Pilih Buat.
6. Setelah AppServertumpukan mencapai status CREATE_COMPLETE, pilih dan pilih tab Output.

   Mungkin Anda perlu menyegarkan beberapa kali untuk memperbarui status.
7. Pada tab Output, catat BucketNamedan InstanceProfileNamenilai untuk digunakan nanti.

> **Note**
>
> AWS CloudFormation menggunakan istilah tumpukan untuk merujuk pada kumpulan sumber daya yang dibuat dari template; itu tidak sama dengan tumpukan AWS OpsWorks Stacks.

Langkah 2: Buat Tumpukan Server Aplikasi PHP

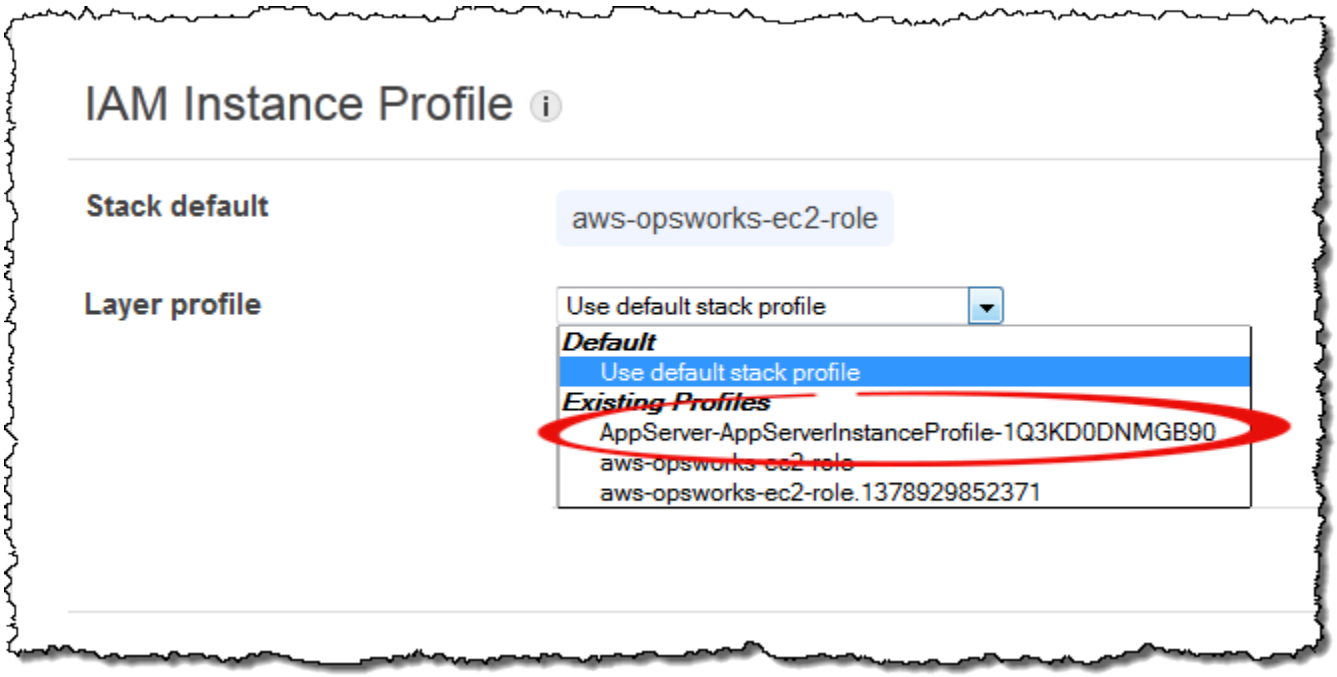
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan terdiri dari dua lapisan, PHP App Server dan MySQL, masing-masing dengan satu instance. Aplikasi ini menyimpan foto di bucket Amazon S3, tetapi menggunakan instance MySQL sebagai penyimpanan data back-end untuk menyimpan metadata untuk setiap foto.

Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Cara Menghapus S3 Bucket? .

Untuk membuat tumpukan

1. Buat tumpukan baru—dinamai **`PhotoSite`** untuk contoh ini—dan tambahkan lapisan PHP App Server. Anda dapat menggunakan pengaturan default untuk keduanya. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru dan Membuat OpsWorks Layer .
2. Pada halaman Layers, untuk PHP App Server, pilih Security dan kemudian pilih Edit.
3. Di bagian Layer Profile, pilih nama profil instance yang Anda rekam sebelumnya, setelah meluncurkan AppServer AWS CloudFormation tumpukan. Ini akan menjadi sesuatu sepertiAppServer-AppServerInstanceProfile-1Q3KD0DNMGB90. AWS OpsWorks Stacks menetapkan profil ini ke semua EC2 instance Amazon layer, yang memberikan izin untuk mengakses bucket Amazon S3 Anda ke aplikasi yang berjalan pada instance layer.



4. Tambahkan instance ke lapisan PHP App Server dan mulai. Untuk informasi selengkapnya tentang cara menambahkan dan memulai instance, lihatMenambahkan Instance ke Layer.
5. Tambahkan layer MySQL ke tumpukan, tambahkan instance, dan mulai. Anda dapat menggunakan pengaturan default untuk layer dan instance. Secara khusus, instance MySQL tidak perlu mengakses bucket Amazon S3, sehingga dapat menggunakan profil instans Stacks AWS OpsWorks standar, yang dipilih secara default.

Langkah 3: Buat dan Men-deploy Buku Masak Kustom

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan belum cukup siap:

- Aplikasi Anda memerlukan beberapa informasi untuk mengakses ke server database MySQL dan bucket Amazon S3, seperti nama host database dan nama bucket Amazon S3.

- Anda perlu menyiapkan database di server database MySQL dan membuat tabel untuk menyimpan metadata foto.

Anda dapat menangani tugas-tugas ini secara manual, tetapi pendekatan yang lebih baik adalah menerapkan resep Chef dan meminta AWS OpsWorks Stacks menjalankan resep secara otomatis pada instance yang sesuai. Resep koki adalah aplikasi Ruby khusus yang digunakan AWS OpsWorks Stacks untuk melakukan tugas pada instance seperti menginstal paket atau membuat file konfigurasi. Mereka dikemas dalam buku masak, yang dapat berisi beberapa resep dan file terkait seperti template untuk file konfigurasi. Buku masak ditempatkan di repositori seperti GitHub, dan harus memiliki struktur direktori standar. Jika Anda belum memiliki repositori buku masak khusus, lihat Repositori informasi tentang cara mengaturnya.

Untuk contoh ini, buku masak telah diimplementasikan untuk Anda dan disimpan dalam GitHub repositori publik. Buku masak berisi dua resep, `appsetup.rb` dan`dbsetup.rb`, dan file templat,`db-connect.php.erb`.

`appsetup.rb`Resep membuat file konfigurasi yang berisi informasi yang dibutuhkan aplikasi untuk mengakses database dan bucket Amazon S3. Ini pada dasarnya adalah versi `appsetup.rb` resep yang dimodifikasi ringan yang dijelaskan dalamConnect Aplikasi ke Database. Perbedaan utama adalah variabel yang diteruskan ke template, yang mewakili informasi akses.

Empat atribut pertama menentukan pengaturan koneksi database, dan secara otomatis ditentukan oleh AWS OpsWorks Stacks ketika Anda membuat instance MySQL.

Ada dua perbedaan antara variabel-variabel ini dan yang ada di resep aslinya:

- Seperti resep asli, `table` variabel mewakili nama tabel database yang dibuat oleh`dbsetup.rb`, dan diatur ke nilai atribut yang didefinisikan dalam file atribut buku masak.

  Namun, atribut memiliki nama yang berbeda:`[:photoapp][:dbtable]`.
- `s3bucket`Variabel khusus untuk contoh ini dan disetel ke nilai atribut yang mewakili nama bucket Amazon S3,. `[:photobucket]`

  `[:photobucket]`didefinisikan dengan menggunakan JSON kustom, seperti yang dijelaskan nanti. Untuk informasi selengkapnya tentang perubahan atribut, lihat Atribut

Untuk informasi selengkapnya tentang atribut, lihatAtribut.

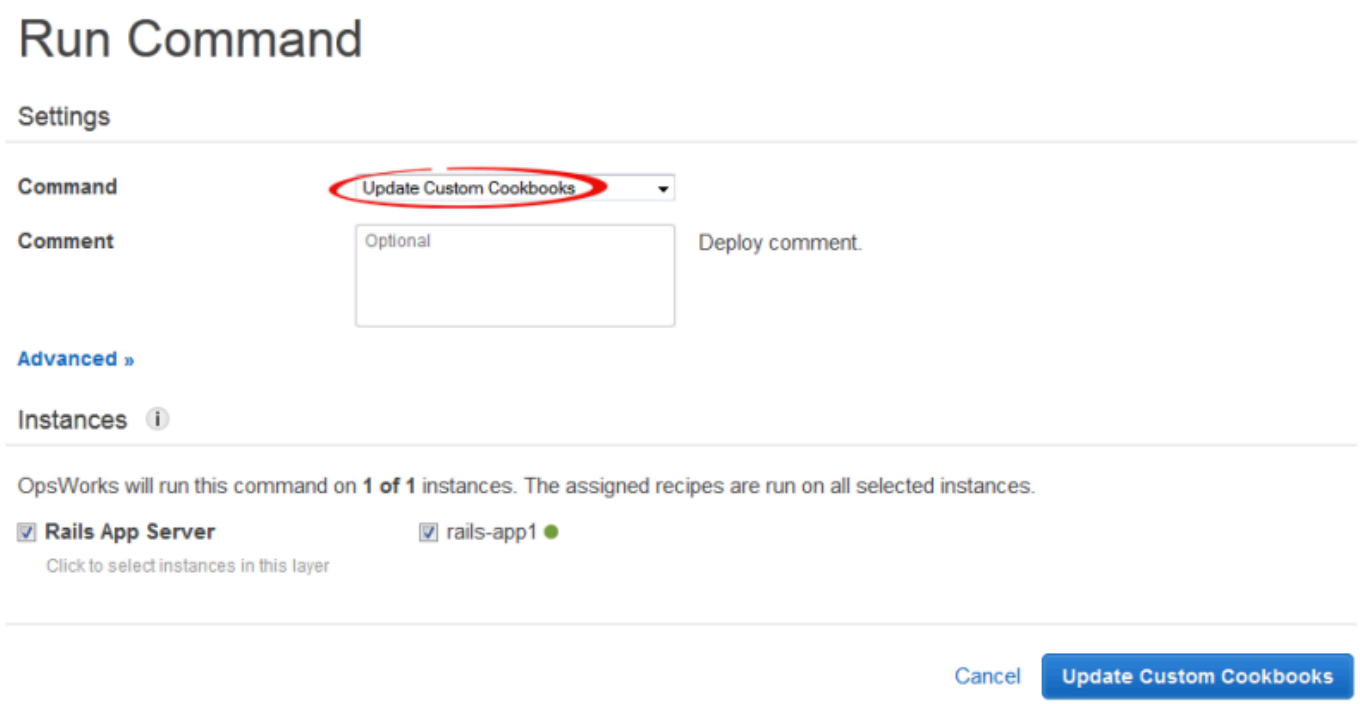
`dbsetup.rb`Resep menyiapkan tabel database untuk menyimpan metadata setiap foto. Ini pada dasarnya adalah versi `dbsetup.rb` resep yang dimodifikasi ringan yang dijelaskan dalamMengatur Database; lihat topik itu untuk deskripsi rinci.

Satu-satunya perbedaan antara contoh ini dan resep asli adalah skema database, yang memiliki tiga kolom yang berisi ID, URL, dan keterangan dari setiap foto yang disimpan di bucket Amazon S3.

Resep sudah diterapkan, jadi yang perlu Anda lakukan hanyalah menyebarkan buku masak photoapp ke cache buku masak setiap instance. AWS OpsWorks Stacks kemudian menjalankan resep cache ketika peristiwa siklus hidup yang sesuai terjadi, seperti yang dijelaskan nanti.

Untuk menerapkan buku masak photoapp

1. Pada halaman AWS OpsWorks Stacks Stack, pilih Stack Settings dan kemudian pilih Edit.
2. Di bagian Manajemen Konfigurasi:
   - Atur Gunakan buku masak Chef khusus ke Ya.
   - Atur jenis Repositori ke Git.
   - Setel URL Repositori ke. **`git://github.com/amazonwebservices/opsworks-example-cookbooks.git`**
3. Pada halaman Stack, pilih Run Command, pilih perintah Update Custom Cookbooks stack, dan kemudian pilih Update Custom Cookbooks untuk menginstal buku masak baru di cache buku masak contoh.



Langkah 4: Tetapkan Resep ke Acara LifeCycle

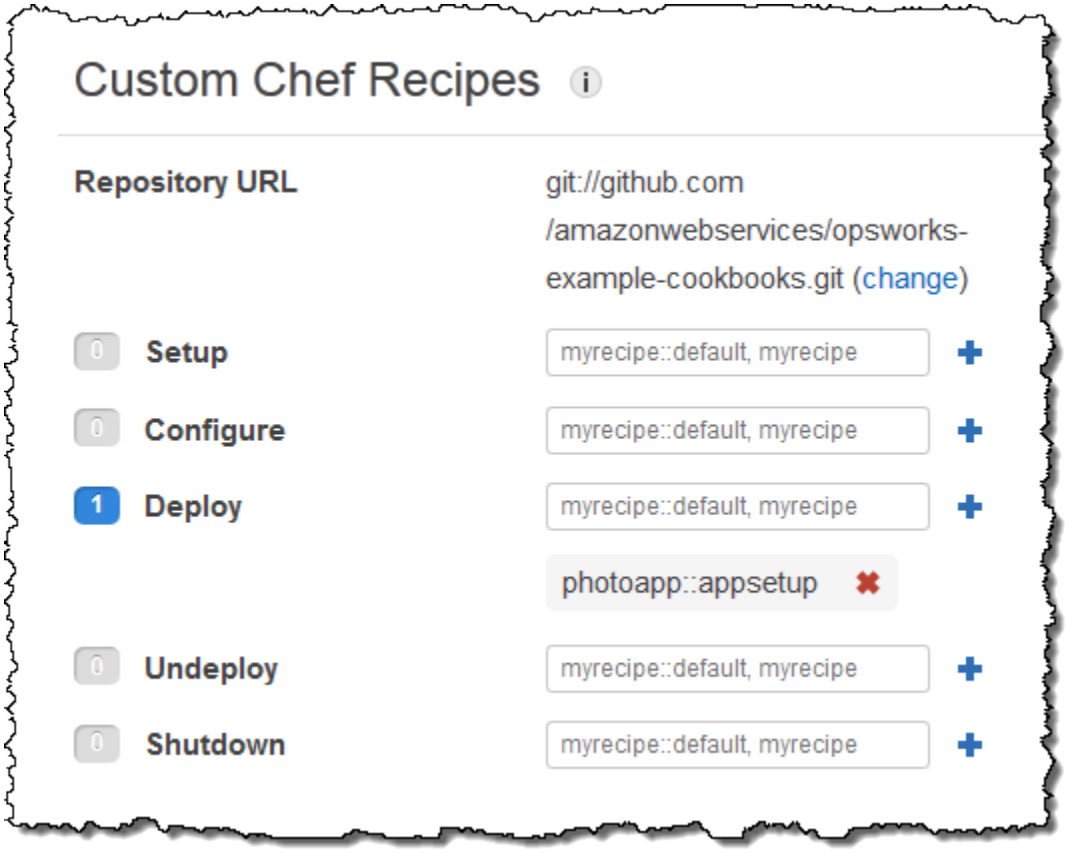
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Anda dapat menjalankan resep khusus secara manual, tetapi pendekatan terbaik biasanya adalah memiliki AWS OpsWorks Stacks menjalankannya secara otomatis. Setiap lapisan memiliki satu set resep bawaan yang ditetapkan untuk masing-masing dari lima peristiwa siklus hidup —Setup, Configure, Deploy, Undeploy, dan Shutdown—. Setiap kali peristiwa terjadi pada sebuah instance, AWS OpsWorks Stacks menjalankan resep terkait untuk setiap lapisan instance, yang menangani tugas yang diperlukan. Misalnya, ketika sebuah instance selesai booting, AWS OpsWorks Stacks memicu peristiwa Setup untuk menjalankan resep Setup, yang biasanya menangani tugas-tugas seperti menginstal dan mengonfigurasi paket.

Anda dapat meminta AWS OpsWorks Stacks menjalankan resep khusus pada instance layer dengan menetapkan setiap resep ke peristiwa siklus hidup yang sesuai. AWS OpsWorks Tumpukan akan menjalankan resep khusus apa pun setelah resep bawaan lapisan selesai. Untuk contoh ini, tetapkan `appsetup.rb` ke event Deploy layer PHP App Server dan `dbsetup.rb` ke event Deploy layer MySQL. AWS OpsWorks Tumpukan kemudian akan menjalankan resep pada instance lapisan terkait selama startup, setelah resep Pengaturan bawaan selesai, dan setiap kali Anda menerapkan aplikasi, setelah resep Deploy yang dibangun selesai. Untuk informasi selengkapnya, lihat Resep Menjalankan Secara Otomatis.

Untuk menetapkan resep kustom ke acara Deploy layer

1. Pada halaman AWS OpsWorks Stacks Layers, untuk PHP App Server pilih Resep dan kemudian pilih Edit.
2. Di bawah Resep Koki Kustom, tambahkan nama resep ke acara penerapan dan pilih +. Nama harus dalam *`cookbookname`*`::`*`recipename`* format Chef, di mana *`recipename`* tidak termasuk `.rb` ekstensi. Untuk contoh ini, Anda masuk`photoapp::appsetup`. Kemudian pilih Simpan untuk memperbarui konfigurasi lapisan.



3. Pada halaman Layers, pilih edit di kolom Actions layer MySQL.

4. Tambahkan `photoapp::dbsetup` ke acara Deploy layer dan simpan konfigurasi baru.

Langkah 5: Tambahkan Informasi Akses ke Konfigurasi Stack dan Atribut Deployment

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

`appsetup.rb`Resepnya bergantung pada data dari konfigurasi tumpukan AWS OpsWorks Stacks dan atribut penerapan, yang diinstal pada setiap instance dan berisi informasi terperinci tentang tumpukan dan aplikasi apa pun yang diterapkan. `deploy`Atribut objek memiliki struktur berikut, yang ditampilkan untuk kenyamanan sebagai JSON:

```
{
   ...
  "deploy": {
    "app1": {
      "application" : "short_name",
      ...
    }
    "app2": {
      ...
    }
    ...
  }
}
```

Node deploy berisi atribut untuk setiap aplikasi yang di-deploy yang diberi nama dengan nama pendek aplikasi. Setiap atribut aplikasi berisi sekumpulan atribut yang menentukan konfigurasi aplikasi, seperti root dokumen dan jenis aplikasi. Untuk daftar `deploy` atribut, lihatmenyebarkan Atribut. Anda dapat mewakili konfigurasi tumpukan dan nilai atribut deployment dalam resep Anda

dengan menggunakan sintaks atribut Chef. Misalnya, `[:deploy][:app1][:application]` mewakili nama pendek aplikasi app1.

Resep kustom bergantung pada beberapa konfigurasi tumpukan dan atribut penerapan yang mewakili database dan informasi akses Amazon S3:

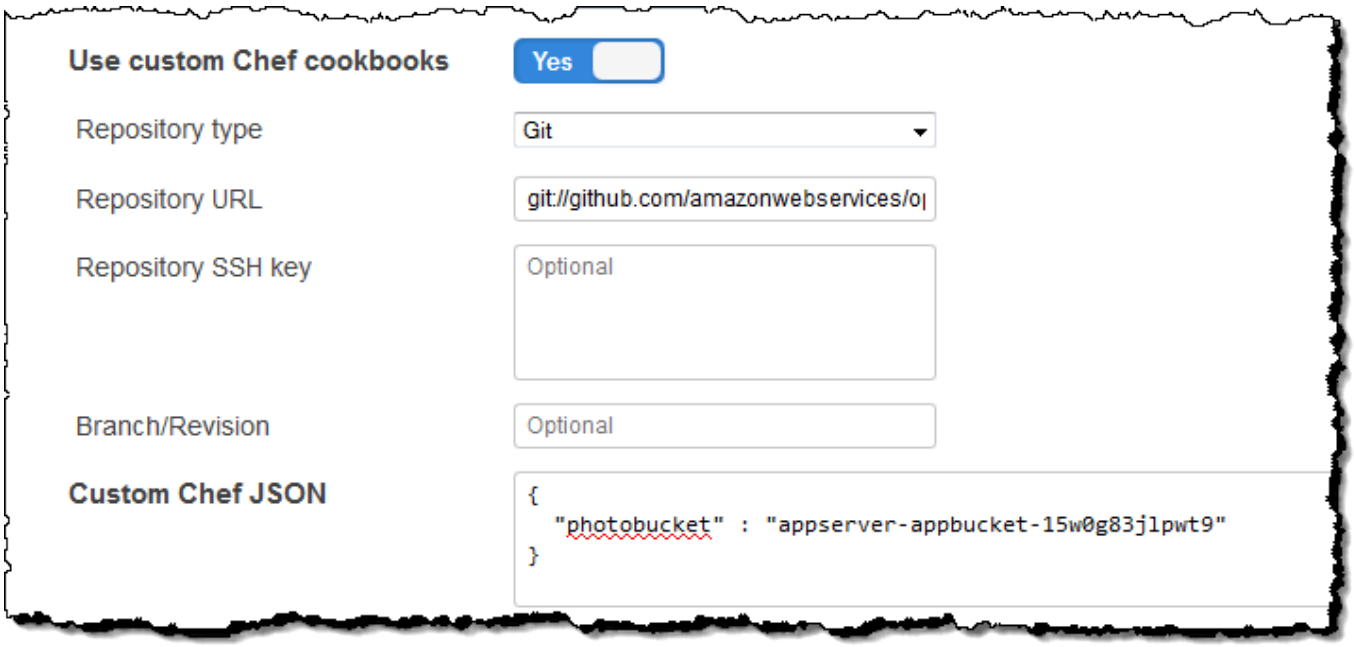
- Atribut koneksi database, seperti`[:deploy][:database][:host]`, didefinisikan oleh AWS OpsWorks Stacks ketika menciptakan lapisan MySQL.
- Atribut nama tabel`[:photoapp][:dbtable]`,, didefinisikan dalam file atribut buku masak kustom, dan diatur `kefoto`.
- Anda harus menentukan atribut nama bucket`[:photobucket]`, dengan menggunakan JSON khusus untuk menambahkan atribut ke konfigurasi tumpukan dan atribut penerapan.

Untuk menentukan atribut nama bucket Amazon S3

1. Pada halaman AWS OpsWorks Stacks Stack, pilih Stack Settings dan kemudian Edit.
2. Di bagian Manajemen Konfigurasi, tambahkan informasi akses ke kotak Custom Chef JSON. Seharusnya terlihat seperti berikut ini:

   ```
   {
     "photobucket" : "yourbucketname"
   }
   ```

   Ganti *yourbucketname* dengan nama bucket yang Anda rekamLangkah 1: Buat Bucket Amazon S3.



AWS OpsWorks Stacks menggabungkan JSON kustom ke dalam konfigurasi tumpukan dan atribut penerapan sebelum menginstalnya pada instance tumpukan; kemudian `appsetup.rb` dapat memperoleh nama bucket dari atribut. `[:photobucket]` Jika Anda ingin mengubah bucket, Anda tidak perlu menyentuh resepnya; Anda bisa mengganti atribut untuk memberikan nama bucket baru.

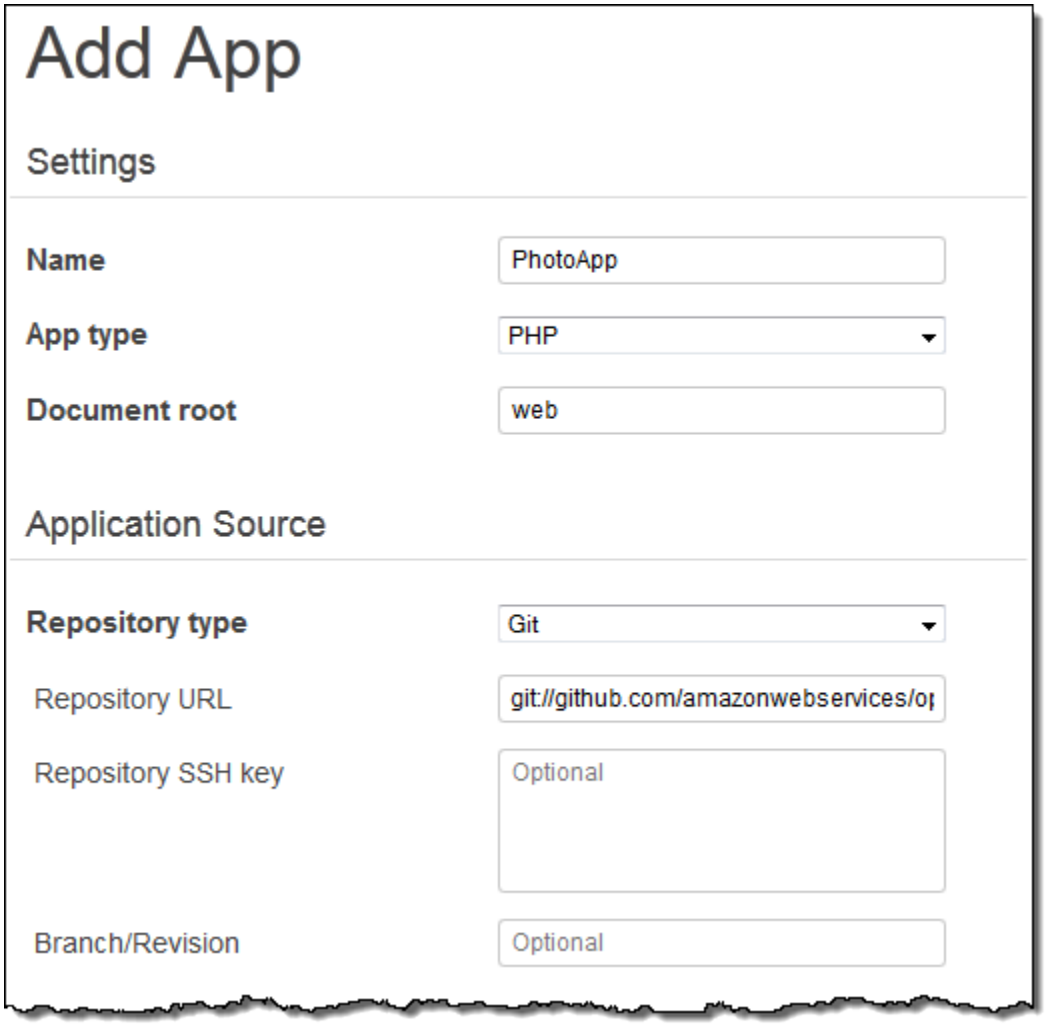
Langkah 6: Men-deploy dan Jalankan PhotoApp

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

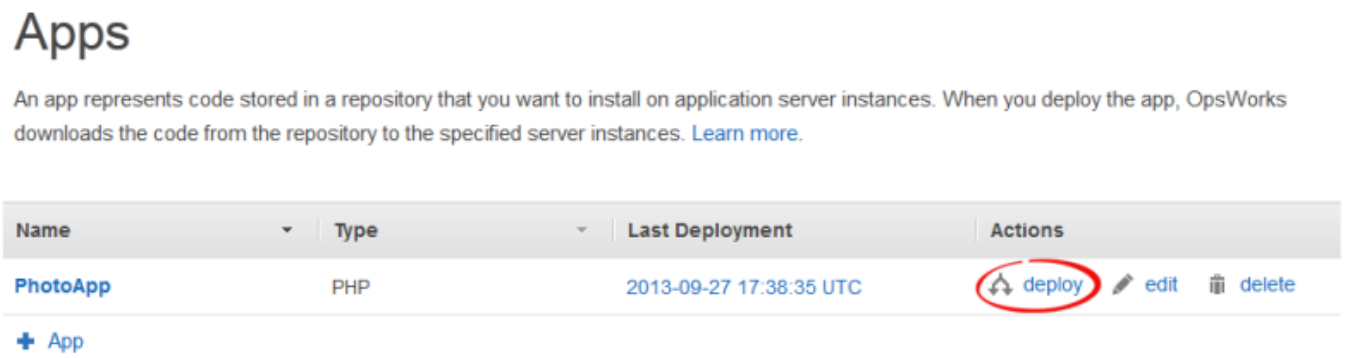
Untuk contoh ini, aplikasi juga telah diterapkan untuk Anda dan disimpan dalam GitHub repositori publik. Anda hanya perlu menambahkan aplikasi ke tumpukan, menyebarkannya ke server aplikasi, dan menjalankannya.

Untuk menambahkan aplikasi ke tumpukan dan menerapkannya ke server aplikasi

1. Buka halaman Aplikasi dan pilih Tambahkan aplikasi.
2. Di halaman Tambahkan Aplikasi, lakukan hal berikut:
   - Setel Nama ke**`PhotoApp`**.
   - Atur jenis Aplikasi ke PHP.
   - Setel root Dokumen ke**`web`**.
   - Atur jenis Repositori ke Git.
   - Setel URL Repositori ke. **`git://github.com/awslabs/opsworks-demo-php-photo-share-app.git`**
   - Pilih Tambah Aplikasi untuk menerima default untuk pengaturan lainnya.

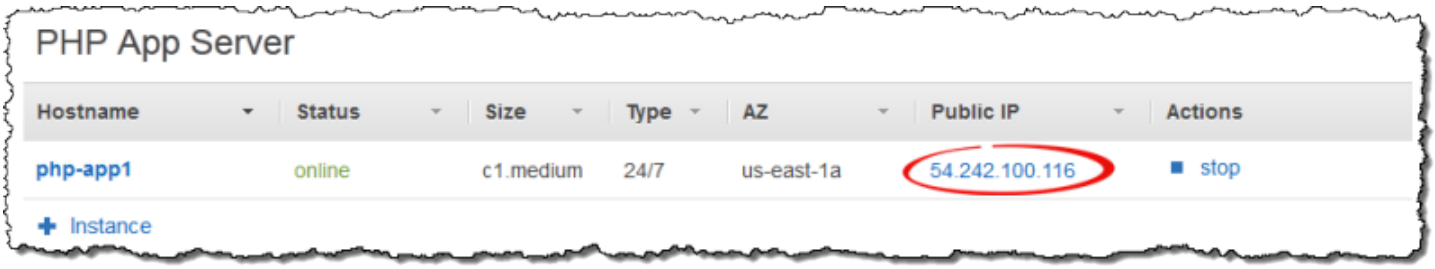


3. Pada halaman Aplikasi, pilih deploy di kolom Tindakan PhotoApp aplikasi.



4. Terima default dan pilih Deploy untuk menyebarkan aplikasi ke server.

Untuk menjalankan PhotoApp, buka halaman Instances dan pilih alamat IP publik instance PHP App Server.



Anda akan melihat antarmuka pengguna berikut. Pilih Tambahkan Foto untuk menyimpan foto di bucket Amazon S3 dan metadata di penyimpanan data back-end.



Menggunakan AWS CodePipeline dengan AWS OpsWorks Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS CodePipelinememungkinkan Anda membuat pipeline pengiriman berkelanjutan yang melacak perubahan kode dari sumber seperti CodeCommit, Amazon Simple Storage Service (Amazon S3), atau. GitHub Anda dapat menggunakannya CodePipeline untuk mengotomatiskan rilis buku masak Chef dan kode aplikasi Anda ke AWS OpsWorks Stacks, di tumpukan Chef 11.10, Chef 12, dan Chef 12.2. Contoh di bagian ini menjelaskan cara membuat dan menggunakan pipeline sederhana dari CodePipeline sebagai alat penyebaran untuk kode yang Anda jalankan di lapisan AWS OpsWorks Stacks.

Note

CodePipeline dan integrasi AWS OpsWorks Stacks tidak didukung untuk diterapkan ke Chef 11.4 dan tumpukan yang lebih lama.

Topik

- AWS CodePipeline dengan AWS OpsWorks Tumpukan - Chef 12 Stacks
- AWS CodePipeline dengan AWS OpsWorks Tumpukan - Chef 11 Stacks

AWS CodePipeline dengan AWS OpsWorks Tumpukan - Chef 12 Stacks

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS CodePipelinememungkinkan Anda membuat pipeline pengiriman berkelanjutan yang melacak perubahan kode dari sumber seperti CodeCommit, Amazon Simple Storage Service (Amazon S3), atau. GitHub Contoh dalam topik ini menjelaskan cara membuat dan menggunakan pipeline sederhana dari CodePipeline sebagai alat penyebaran untuk kode yang Anda jalankan di lapisan

AWS OpsWorks Stacks. Dalam contoh ini, Anda membuat pipeline untuk aplikasi Node.js sederhana, lalu menginstruksikan AWS OpsWorks Stacks untuk menjalankan aplikasi pada semua instance dalam lapisan dalam tumpukan Chef 12 (dalam hal ini, satu instance).

> **Note**
>
> Topik ini menjelaskan cara menggunakan pipeline untuk menjalankan dan memperbarui aplikasi di tumpukan Chef 12. Untuk informasi tentang cara menggunakan pipeline untuk menjalankan dan memperbarui aplikasi di tumpukan Chef 11.10, lihatAWS CodePipeline dengan AWS OpsWorks Tumpukan - Chef 11 Stacks. Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3? atau Bagaimana Cara Menghapus Bucket S3? .

Topik

- Prasyarat
- Skenario lainnya yang didukung
- Langkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks
- Langkah 2: Konfigurasikan tumpukan dan lapisan Anda agar bisa menggunakan buku masak khusus
- Langkah 3: Unggah kode aplikasi ke bucket Amazon S3
- Langkah 4: Tambahkan aplikasi Anda ke AWS OpsWorks Stacks
- Langkah 5: Buat alur di CodePipeline
- Langkah 6: Memverifikasi penerapan aplikasi di Stacks AWS OpsWorks
- Langkah 7 (Opsional): Perbarui kode aplikasi untuk melihat CodePipeline penerapan ulang aplikasi Anda secara otomatis
- Langkah 8 (Opsional): Bersihkan Sumber Daya

Prasyarat

Sebelum Anda memulai panduan ini, pastikan Anda memiliki izin administrator untuk melakukan semua tugas berikut. Anda dapat menjadi anggota grup yang menerapkan AdministratorAccesskebijakan tersebut, atau Anda dapat menjadi anggota grup yang memiliki izin dan kebijakan yang ditampilkan dalam tabel berikut. Sebagai praktik keamanan terbaik, Anda harus

menjadi anggota grup yang memiliki izin untuk melakukan tugas berikut, alih-alih menetapkan izin yang diperlukan untuk pengguna individual.

Untuk informasi selengkapnya tentang membuat grup keamanan di IAM dan menetapkan izin ke grup, lihat Membuat grup pengguna IAM. Untuk informasi selengkapnya tentang mengelola izin AWS OpsWorks Tumpukan, lihat Praktik Terbaik: Mengelola Izin.

Izin	Kebijakan yang Direkomendasikan untuk Dilampirkan ke Grup
Buat dan edit tumpukan, lapisan, dan instance di AWS OpsWorks Stacks.	AWSOpsWorks_FullAccess
Buat, edit, dan jalankan template di AWS CloudFormation.	AmazonCloudFormationFullAccess
Membuat, mengedit, dan mengakses bucket Amazon S3.	AmazonS3 FullAccess
Buat, edit, dan jalankan pipeline CodePipeline, terutama pipeline yang menggunakan AWS OpsWorks Stacks sebagai penyedia.	AWSCodePipeline_FullAccess

Anda juga harus memiliki Amazon EC2 key pair. Anda akan diminta untuk memberikan nama key pair ini ketika Anda menjalankan AWS CloudFormation template yang membuat sample stack, layer, dan instance dalam panduan ini. Untuk informasi selengkapnya tentang mendapatkan key pair di EC2 konsol Amazon, lihat Membuat Pasangan Kunci di EC2 dokumentasi Amazon. Pasangan kunci harus berada di Wilayah US East (N. Virginia). Anda dapat menggunakan pasangan kunci yang sudah ada jika Anda sudah memilikinya di wilayah tersebut.

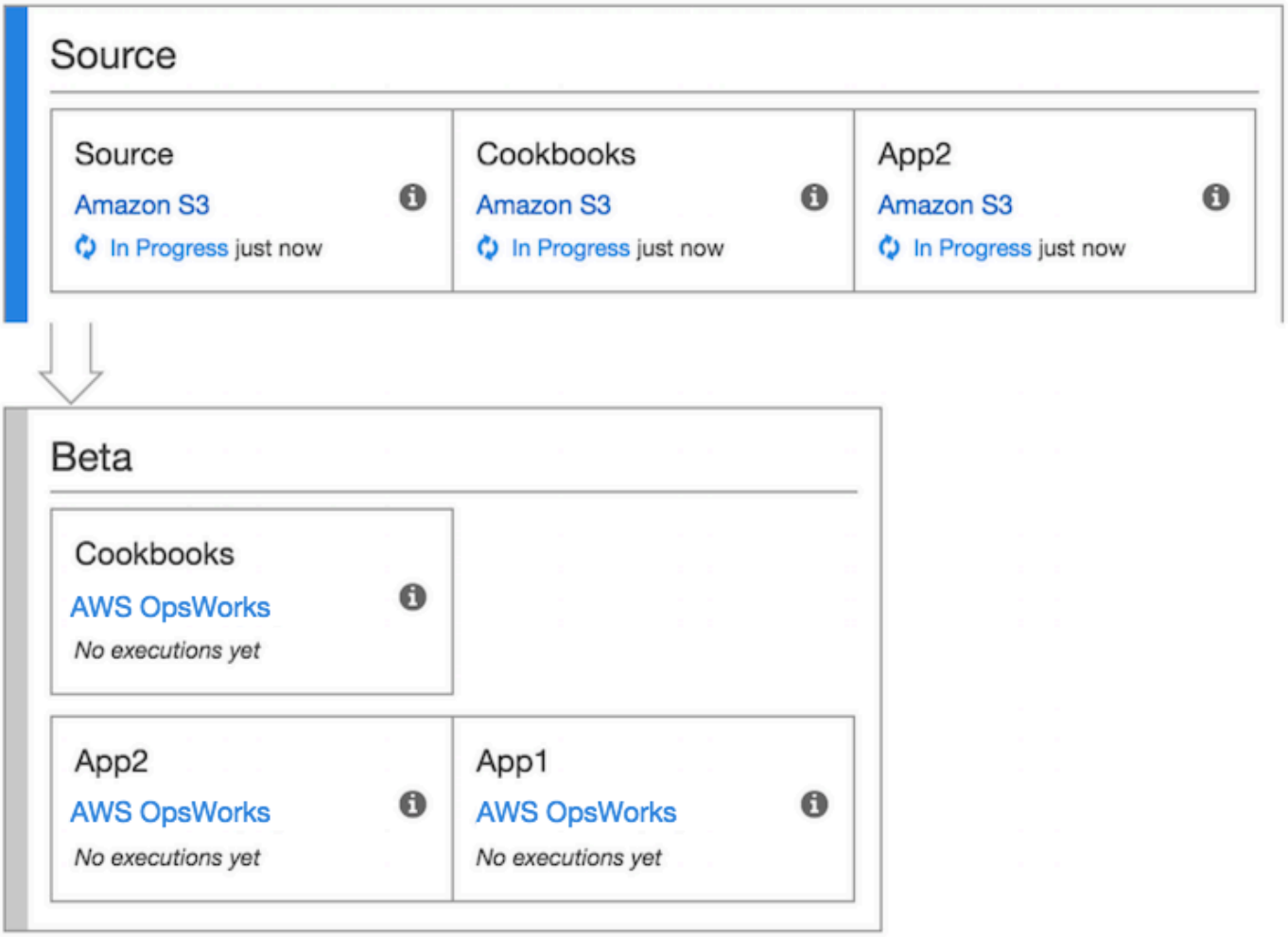
Skenario lainnya yang didukung

Panduan ini membuat pipeline sederhana yang mencakup satu tahap Sumber dan satu Penerapan. Namun, Anda dapat membuat pipeline yang lebih kompleks yang menggunakan AWS OpsWorks Stacks sebagai penyedia. Berikut ini adalah contoh pipa dan skenario yang didukung:

- Anda dapat mengedit pipeline untuk menambahkan buku masak Chef ke tahap Sumber dan target terkait untuk buku masak yang diperbarui ke tahap Deploy. Dalam hal ini, Anda menambahkan

tindakan Deploy yang memicu pembaruan buku masak Anda saat Anda membuat perubahan pada sumbernya. Buku masak yang diperbarui diterapkan sebelum aplikasi Anda.

- Anda dapat membuat pipeline yang kompleks, dengan buku masak khusus dan beberapa aplikasi, dan menerapkan ke AWS OpsWorks tumpukan Stacks. Pipeline melacak perubahan pada sumber aplikasi dan buku masak, dan digunakan kembali ketika Anda telah membuat perubahan. Berikut ini adalah contoh alur yang serupa dan kompleks:



Untuk informasi selengkapnya tentang bekerja dengan CodePipeline, lihat Panduan CodePipeline Pengguna.

Langkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk menggunakan AWS OpsWorks Stacks sebagai penyedia penyebaran untuk pipeline, Anda harus terlebih dahulu memiliki tumpukan, lapisan, dan setidaknya satu instance di lapisan. Meskipun Anda dapat membuat tumpukan di AWS OpsWorks Stacks dengan mengikuti petunjuk di Memulai dengan Tumpukan Linux atau Memulai dengan Tumpukan Windows, untuk menghemat waktu Anda, contoh ini menggunakan AWS CloudFormation templat untuk membuat tumpukan, lapisan, dan instance Chef 12 berbasis Linux. Instance yang dibuat oleh template ini menjalankan Amazon Linux 2016.03, dan memiliki tipe instance. `c3.large` Meskipun template tidak mengonfigurasi tumpukan Anda untuk menggunakan buku masak khusus, Anda akan melakukannya nanti di penelusuran.

> **Important**
>
> AWS CloudFormation Template harus disimpan dan dijalankan di wilayah yang sama dengan bucket Amazon S3 tempat Anda kemudian mengunggah aplikasi dan wilayah yang sama tempat Anda nantinya membuat pipeline. CodePipeline Saat ini, CodePipeline mendukung penyedia AWS OpsWorks Stacks di Wilayah US East (N. Virginia) (us-east-1). Semua sumber daya dalam panduan ini harus dibuat di Wilayah US East (N. Virginia).
> Jika pembuatan tumpukan gagal, Anda mungkin mendekati jumlah maksimum peran IAM yang diizinkan untuk akun Anda. Pembuatan tumpukan juga bisa gagal jika akun Anda tidak dapat meluncurkan `c3.large` instance dengan tipe instans. Misalnya, jika Anda menggunakan Tingkat AWS Gratis, Anda mungkin menerima kesalahan sepertiRoot `device type: must be included in EBS`. Jika akun Anda memiliki batasan pada jenis instans yang diizinkan untuk dibuat, seperti batasan yang diberlakukan oleh Tingkat AWS Gratis, coba ubah nilai `InstanceType` parameter di blok instans templat menjadi jenis instans yang dapat digunakan akun Anda.

Untuk membuat stack, layer, dan instance menggunakan AWS CloudFormation

1. Salin AWS CloudFormation template berikut ke dalam dokumen teks biasa baru. Simpan file ke lokasi yang nyaman di komputer lokal Anda, dan beri nama NewOpsWorksStack.template, atau nama lain yang nyaman bagi Anda.

```
{
  "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",
```

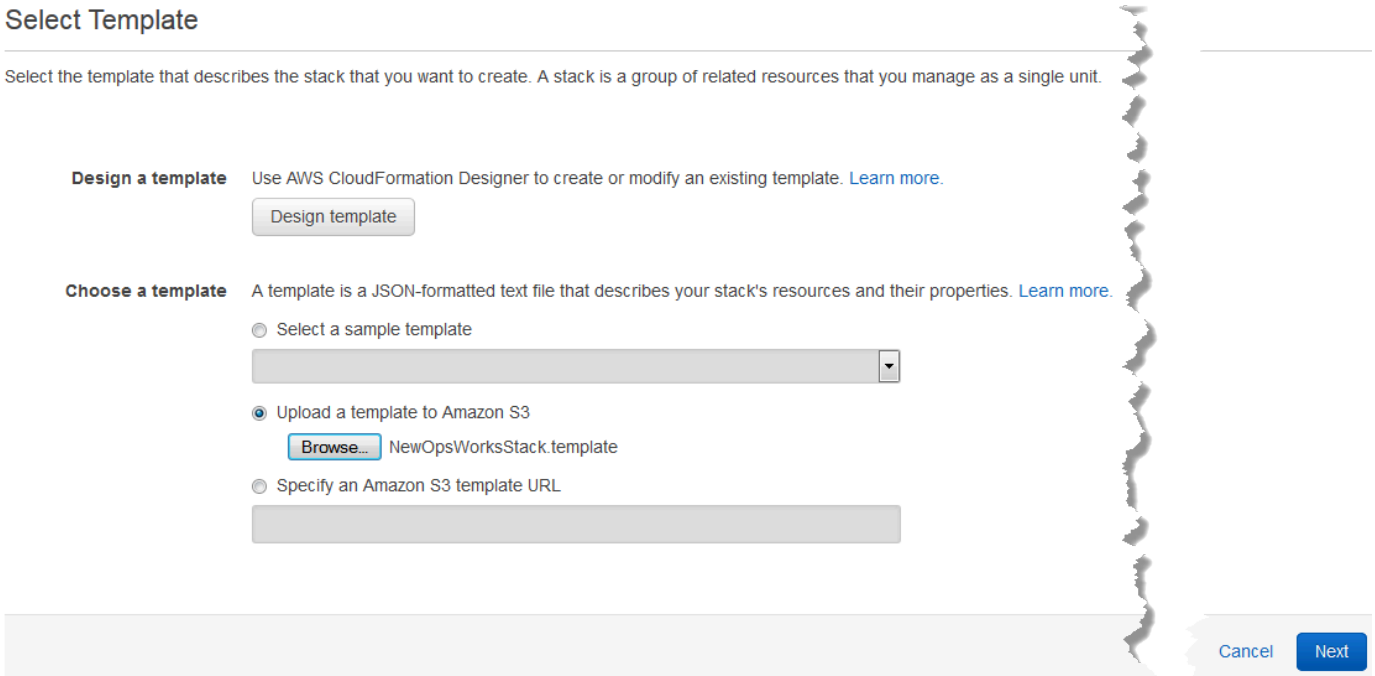
```
"EnableAutoHealing": "true",
      "AutoAssignElasticIps": "false",
      "AutoAssignPublicIps": "true",
"CustomSecurityGroupIds": [
  {
    "Fn::GetAtt": [
           "CPOpsDeploySecGroup", "GroupId"
    ]
  }
 ]
    },
    "DependsOn": [
      "MyStack",
      "CPOpsDeploySecGroup"
    ]
  },
  "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"
      }
    }
  }
}
```

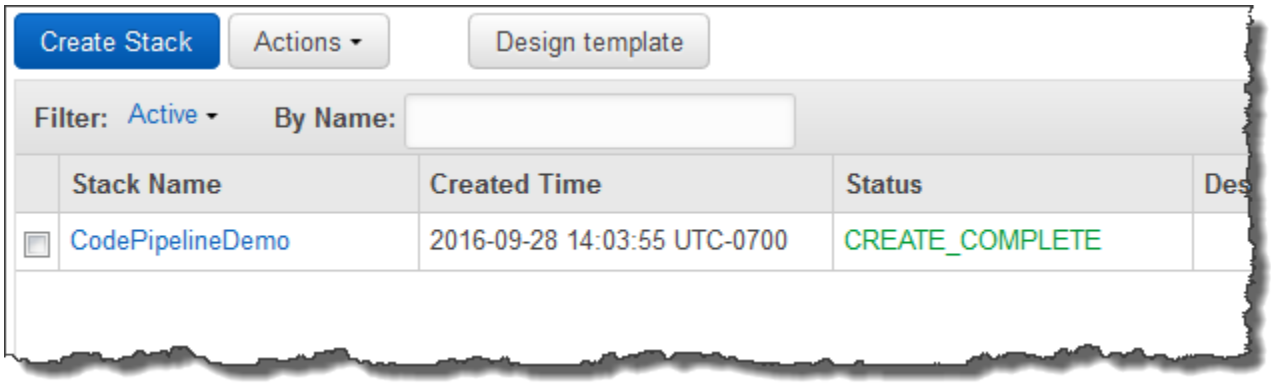
2. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
3. Di AWS CloudFormation halaman beranda, pilih Buat tumpukan.
4. Pada halaman Pilih Templat, di area Pilih templat, pilih Unggah templat ke Amazon S3, lalu pilih Jelajahi.
5. Jelajahi AWS CloudFormation template yang Anda simpan di langkah 1, lalu pilih Buka. Pada halaman Pilih Templat, pilih Selanjutnya.



6. Pada halaman Tentukan Detail, beri nama tumpukan CodePipelineDemo, atau nama tumpukan apa pun yang unik untuk akun Anda. Jika Anda memilih nama yang berbeda untuk tumpukan Anda, ubah nama tumpukan di seluruh panduan ini.
7. Di area Parameter, berikan nama EC2 key pair yang ingin Anda gunakan untuk mengakses instance AWS OpsWorks Stacks Anda setelah dibuat. Pilih Berikutnya.
8. Pada halaman Opsi, pilih Selanjutnya. (Pengaturan di halaman ini tidak diperlukan untuk panduan ini.)
9. AWS CloudFormation Template yang Anda gunakan dalam panduan ini menciptakan peran IAM, profil instance, dan instance.

   > **Important**
   >
   > Sebelum Anda memilih Buat, pilih Biaya untuk memperkirakan biaya yang mungkin dikenakan AWS untuk membuat sumber daya dengan templat ini.

   Jika membuat sumber daya IAM dapat diterima, pilih kotak centang Saya mengakui bahwa templat ini dapat AWS CloudFormation menyebabkan membuat sumber daya IAM, lalu pilih Buat. Jika membuat sumber daya IAM tidak dapat diterima, Anda tidak dapat melanjutkan prosedur ini.
10. Di AWS CloudFormation dasbor, Anda dapat melihat kemajuan pembuatan tumpukan. Sebelum Anda melanjutkan ke langkah berikutnya, tunggu hingga CREATE_COMPLETE ditampilkan di kolom Status.



Untuk memverifikasi pembuatan tumpukan di AWS OpsWorks Stacks

1. Buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.

2. Di dasbor AWS OpsWorks Stacks, lihat tumpukan yang Anda buat.



3. Buka tumpukan, dan lihat layer dan instance. Perhatikan bahwa layer dan instance dibuat dengan nama dan metadata lain yang disediakan dalam template. AWS CloudFormation Anda siap mengonfigurasi tumpukan dan lapisan Anda untuk menggunakan buku masak dan resep Chef khusus.

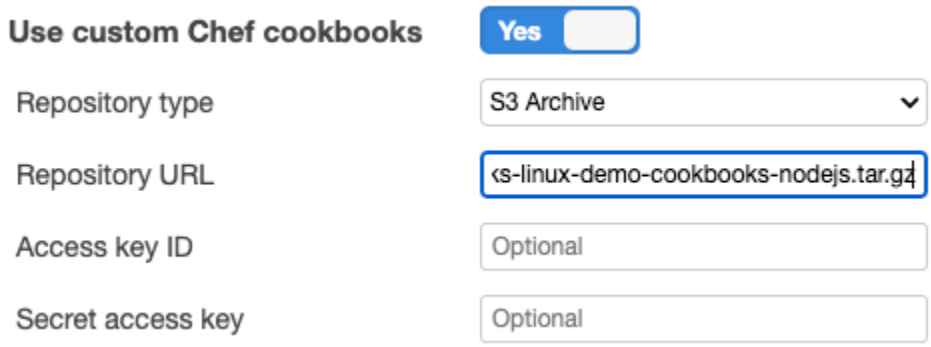
Langkah 2: Konfigurasikan tumpukan dan lapisan Anda agar bisa menggunakan buku masak khusus

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Tumpukan Chef 12 di AWS OpsWorks Stacks memerlukan buku masak Anda sendiri atau buatan komunitas untuk membangun lapisan aplikasi khusus. Untuk panduan ini, Anda dapat menunjuk ke repositori yang berisi satu set buku masak Chef dan resep Chef. Resep-resep ini menginstal paket Node.js dan dependensinya pada instance Anda. Anda akan menggunakan resep Chef lain untuk menyebarkan aplikasi Node.js yang akan Anda siapkan. Langkah 4: Tambahkan aplikasi Anda ke AWS OpsWorks Stacks Resep Chef yang Anda tentukan dalam langkah ini berjalan setiap kali versi baru aplikasi Anda digunakan oleh CodePipeline.

1. Di konsol AWS OpsWorks Stacks, buka tumpukan yang Anda buat. Langkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks Pilih Pengaturan Stack, lalu pilih Edit.
2. Atur Gunakan buku masak Chef khusus ke Ya. Ini menunjukkan pengaturan buku masak khusus terkait.
3. Dari daftar drop-down tipe Repositori, pilih Arsip S3. Untuk bekerja dengan keduanya CodePipeline dan AWS OpsWorks, sumber buku masak Anda harus S3.

4. Untuk URL Repositori, tentukan. **`https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz`** Pengaturan Anda akan menyerupai hal-hal berikut ini.



5. Pilih Simpan.
6. Di panel navigasi, pilih Layers (Lapisan).
7. Pilih Pengaturan untuk layer yang Anda buatLangkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks.
8. Pada tab General Settings, pastikan bahwa nama layer adalah Node.js App Server, dan nama pendek layer adalah app1. Pilih Resep.
9. Pada tab Resep, tentukan resep **`nodejs_demo`** yang ingin Anda jalankan selama acara siklus hidup Deploy. Pilih Simpan.
10. Pada tab Keamanan, dari daftar drop-down Grup keamanan, pilih grup keamanan AWS-OpsWorks -Webapp.
11. Pilih Simpan.

Langkah 3: Unggah kode aplikasi ke bucket Amazon S3

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

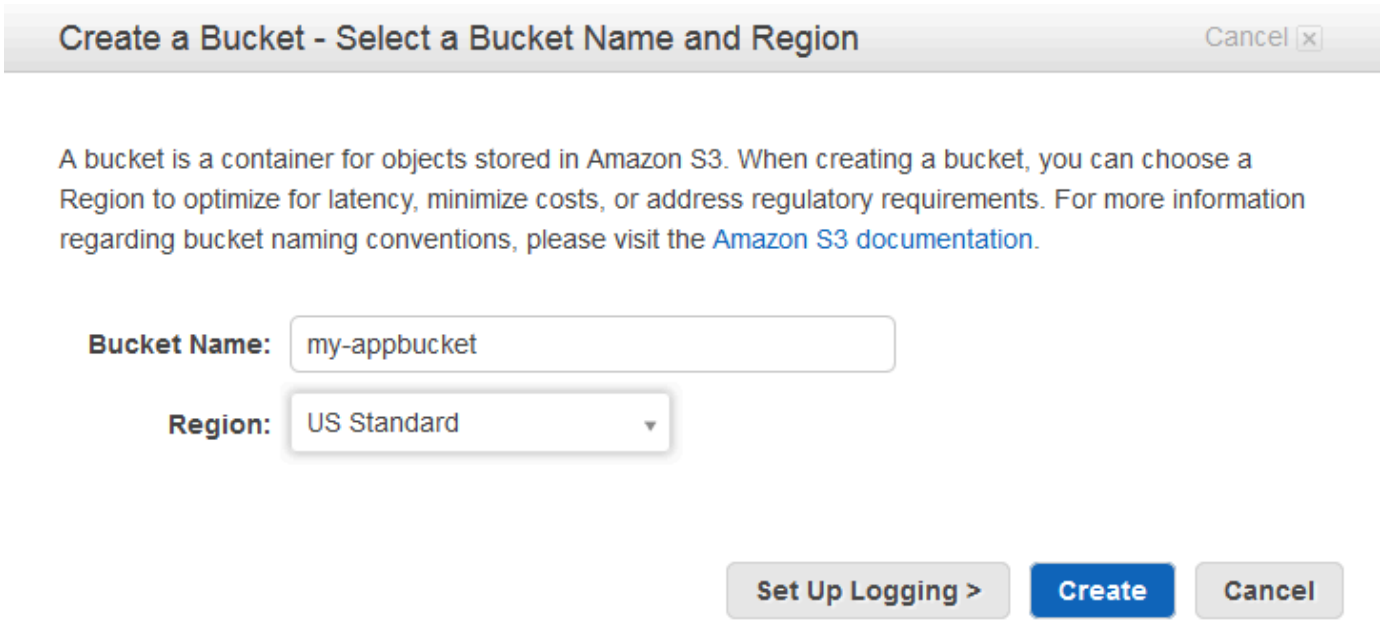
Karena Anda harus menyediakan tautan ke repositori kode Anda sebagai bagian dari penyiapan pipeline, siapkan repositori kode sebelum membuat pipeline. Di panduan ini, Anda mengunggah aplikasi Node.js ke bucket Amazon S3.

Meskipun CodePipeline dapat menggunakan kode langsung dari GitHub atau CodeCommit sebagai sumber, panduan ini menunjukkan cara menggunakan bucket Amazon S3. Dalam panduan ini, Anda mengunggah contoh aplikasi Node.js ke bucket Amazon S3 Anda sendiri, sehingga Anda dapat membuat perubahan pada aplikasi. Bucket Amazon S3 yang Anda buat pada langkah ini memungkinkan CodePipeline untuk mendeteksi perubahan pada kode aplikasi dan menerapkan aplikasi yang diubah secara otomatis. Jika mau, Anda dapat menggunakan bucket yang sudah ada. Pastikan bucket memenuhi kriteria yang dijelaskan dalam Simple Pipeline Walkthrough (Amazon S3 Bucket) dalam dokumentasi. CodePipeline

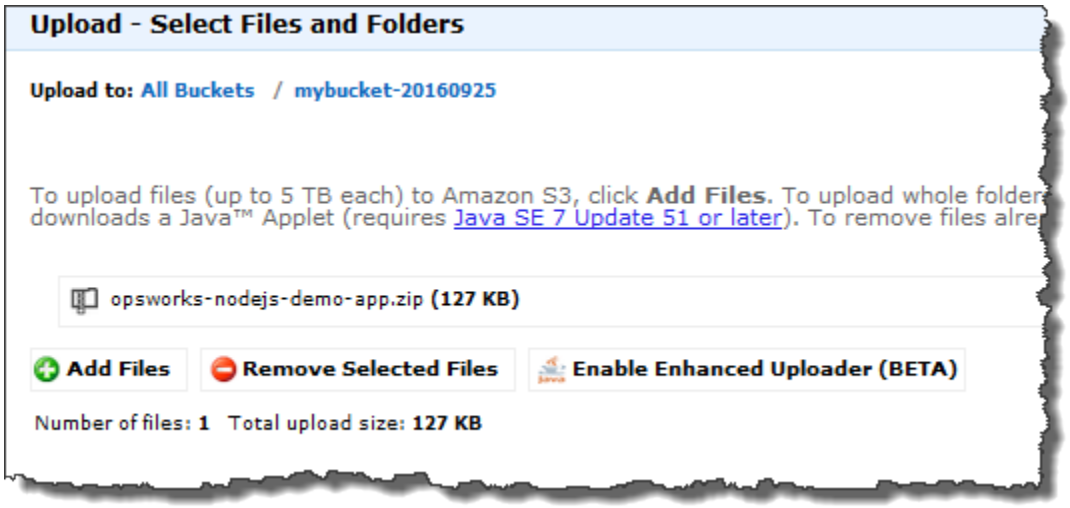
> **Important**
>
> Bucket Amazon S3 harus berada di wilayah yang sama dengan tempat Anda nantinya akan membuat alur. Saat ini, CodePipeline mendukung penyedia AWS OpsWorks Stacks di Wilayah US East (N. Virginia) (us-east-1). Semua sumber daya dalam panduan ini harus dibuat di Wilayah US East (N. Virginia). Bucket juga harus berversi karena CodePipeline memerlukan sumber berversi. Untuk informasi lebih lanjut, lihat Penggunaan Versioning.

Untuk mengunggah aplikasi Anda ke bucket Amazon S3

1. Unduh file ZIP dari sampel AWS OpsWorks Stacks, aplikasi Node.js, dan simpan ke lokasi yang nyaman di komputer lokal Anda.
2. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
3. Pilih Buat Bucket.
4. Pada halaman Buat Bucket - Pilih Nama Bucket dan Wilayah, untuk Nama Bucket, ketikkan nama unik untuk bucket Anda. Nama bucket harus unik di semua AWS akun, tidak hanya di akun Anda sendiri. Panduan ini menggunakan nama**`my-appbucket`**, tetapi Anda dapat menggunakannya `my-appbucket-`*`yearmonthday`* untuk membuat nama ember Anda unik. Dari daftar drop-down Region, pilih Standar AS, lalu pilih Buat. Standar AS setara dengan`us-east-1`.



5. Pilih bucket yang Anda buat dari daftar Semua Bucket.
6. Pada halaman bucket, pilih Upload.
7. Pada halaman Unggah - Pilih File dan Folder, pilih Tambahkan file. Jelajahi file ZIP yang Anda simpan di langkah 1, pilih Buka, lalu pilih Mulai Unggah.



8. Setelah unggahan selesai, pilih file ZIP dari daftar file di bucket Anda, lalu pilih Properties.
9. Di panel Properties, salin tautan ke file ZIP Anda, dan catat tautannya. Anda akan memerlukan nama bucket dan bagian nama file ZIP dari tautan ini untuk membuat pipeline Anda.

Langkah 4: Tambahkan aplikasi Anda ke AWS OpsWorks Stacks

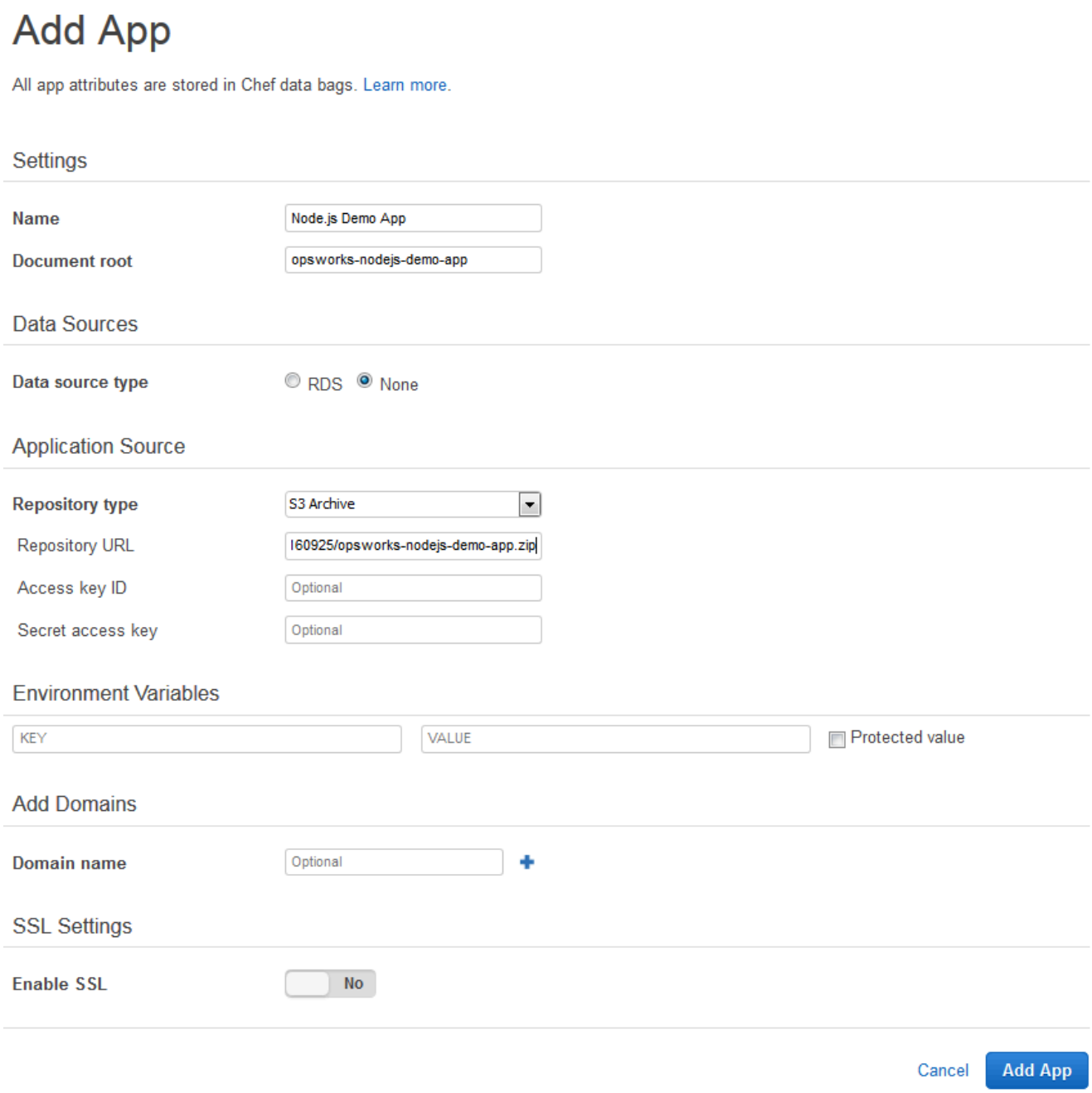
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sebelum Anda membuat pipeline CodePipeline, tambahkan aplikasi pengujian Node.js ke AWS OpsWorks Stacks. Saat membuat pipeline, Anda harus memilih aplikasi yang telah ditambahkan ke AWS OpsWorks Stacks.

Siapkan tautan bucket Amazon S3 dari langkah 9 dari prosedur sebelumnya. Anda akan memerlukan tautan ke bucket tempat Anda menyimpan aplikasi pengujian untuk menyelesaikan prosedur ini.

Untuk menambahkan aplikasi ke AWS OpsWorks Stacks

1. Di konsol AWS OpsWorks Stacks, buka CodePipelineDemo, dan di panel navigasi, pilih Aplikasi.
2. Pilih Tambahkan aplikasi.
3. Di halaman Tambahkan Aplikasi, berikan informasi berikut:
   a. Tentukan nama untuk aplikasi Anda. Panduan ini menggunakan nama. `Node.js Demo App`
   b. Untuk tipe sumber data, pilih Tidak ada. Aplikasi ini tidak memerlukan database eksternal atau sumber data.
   c. Dalam daftar drop-down Jenis repositori, pilih Arsip S3.
   d. Di kotak string URL Repositori, tempel URL yang Anda salin di langkah 9 dari. Langkah 3: Unggah kode aplikasi ke bucket Amazon S3 Formulir Anda harus serupa dengan berikut ini:



4. Anda tidak perlu mengubah pengaturan lainnya di formulir ini. Pilih Tambahkan Aplikasi.
5. Saat aplikasi Aplikasi Demo Node.js muncul di daftar di halaman Aplikasi, lanjutkan ke prosedur berikutnyaLangkah 5: Buat alur di CodePipeline.

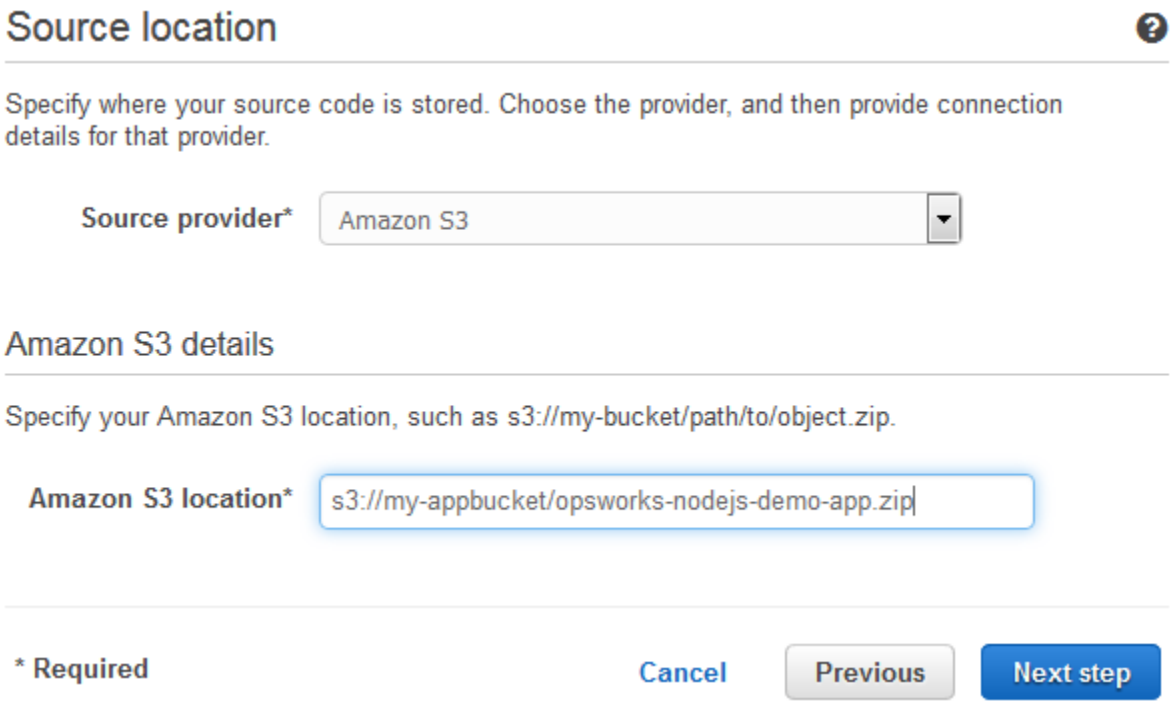
Langkah 5: Buat alur di CodePipeline

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

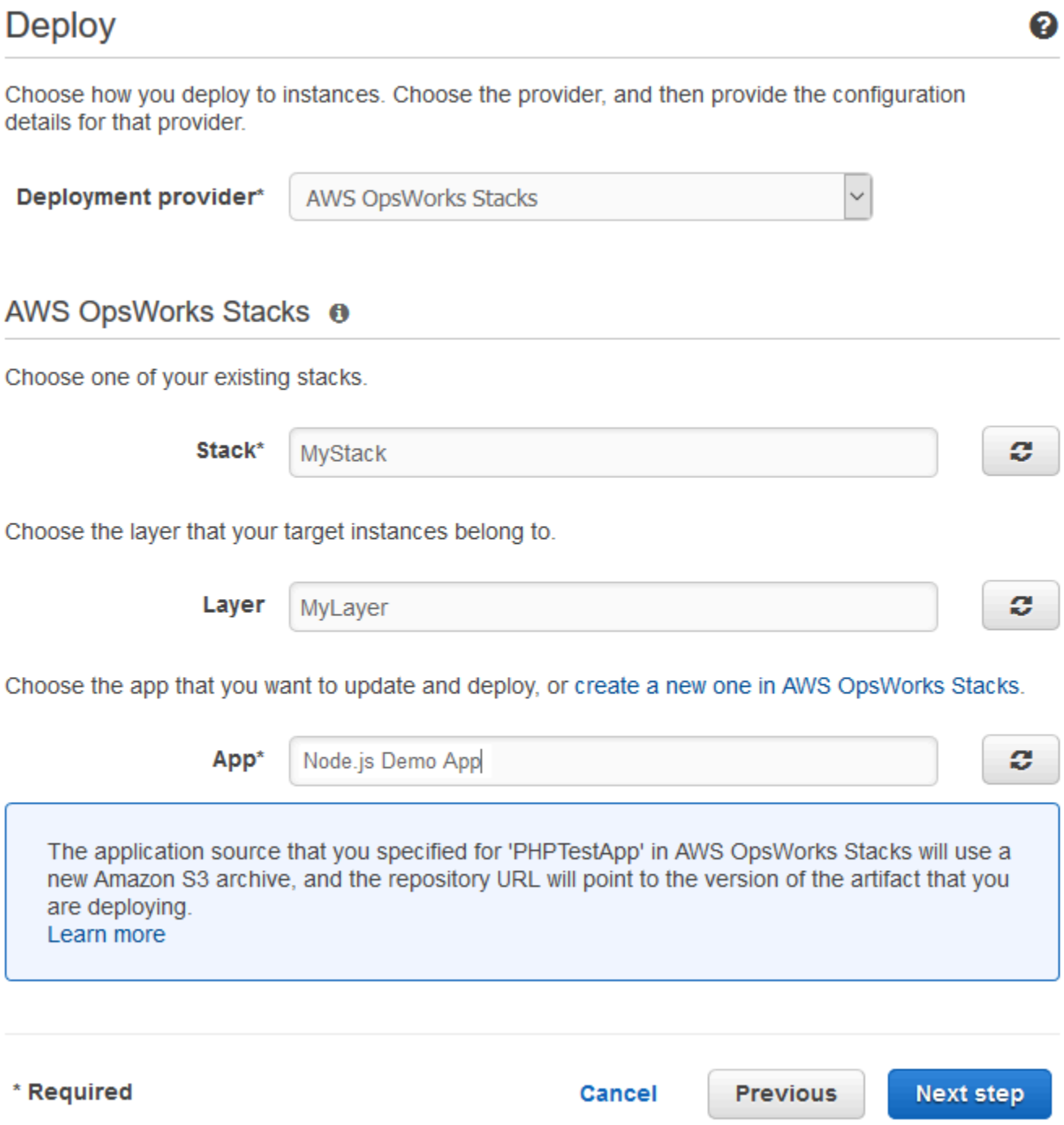
Setelah Anda memiliki tumpukan dengan lapisan dan setidaknya satu instance yang dikonfigurasi di AWS OpsWorks Stacks, buat pipeline CodePipeline dengan AWS OpsWorks Stacks sebagai penyedia untuk menyebarkan aplikasi atau buku masak Chef ke sumber daya Stacks Anda. AWS OpsWorks

Untuk membuat alur

1. Buka CodePipeline konsol di https://console.aws.amazon.com/codepipeline/.
2. Pilih Buat pipeline.
3. Pada CodePipeline halaman Memulai dengan**`MyOpsWorksPipeline`**, ketik, atau nama pipeline lainnya yang unik untuk akun Anda, lalu pilih Langkah berikutnya.
4. Pada halaman Lokasi Sumber, pilih Amazon S3 dari daftar drop-down penyedia Sumber.
5. Di area detail Amazon S3, ketik jalur bucket Amazon S3 Anda, dalam format. **`s3://`*`bucket-name`*`/`*`file name`* Lihat tautan yang Anda catat di langkah 9 dariLangkah 3: Unggah kode aplikasi ke bucket Amazon S3. Dalam panduan ini, jalannya adalah. `s3://my-appbucket/opsworks-nodejs-demo-app.zip` Pilih Langkah selanjutnya.



6. Pada halaman Build, pilih No Build dari daftar drop-down, lalu pilih Langkah berikutnya.
7. Pada halaman Deploy, pilih AWS OpsWorks Stacks sebagai penyedia penyebaran.



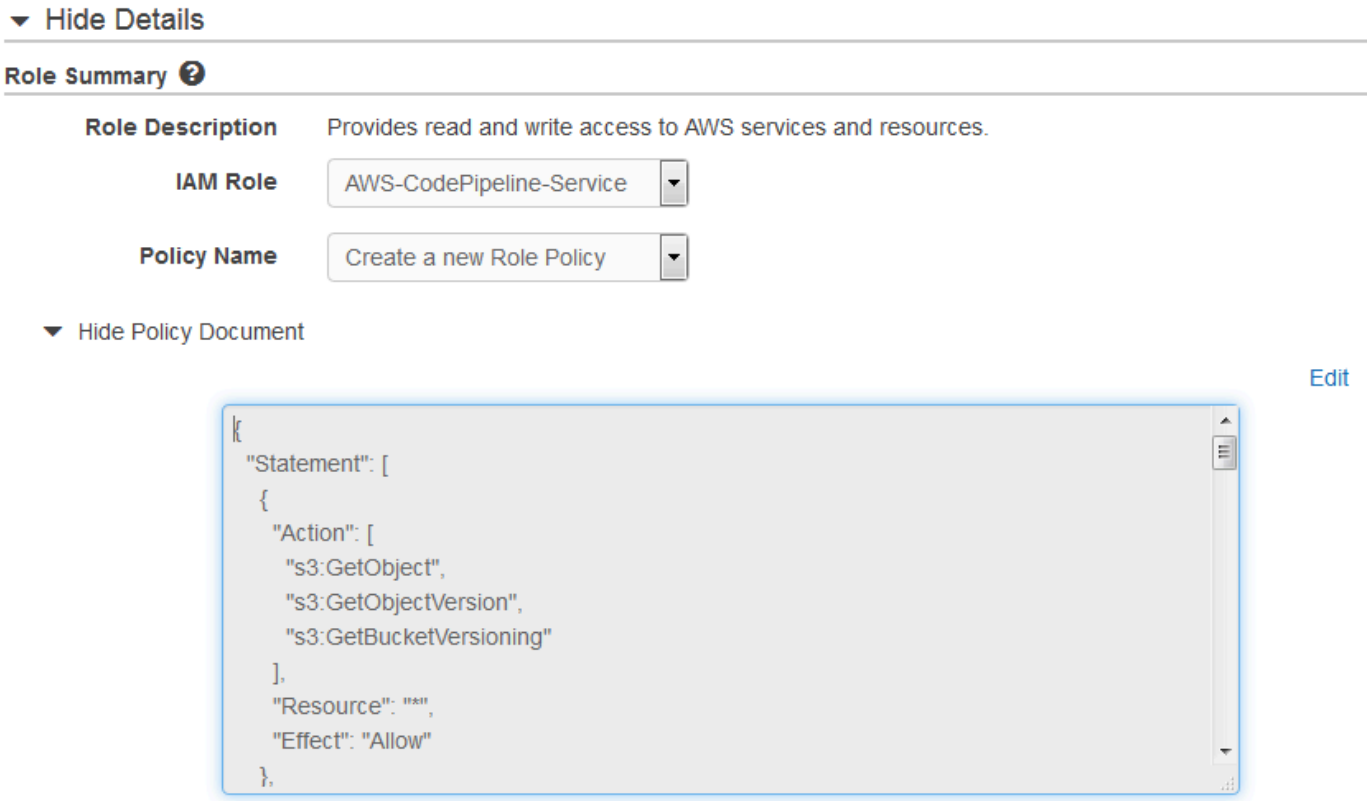
8. Di bidang Stack`CodePipelineDemo`, ketik, atau nama tumpukan yang Anda buatLangkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks.
9. Di bidang Layer`Node.js App Server`, ketik, atau nama layer yang Anda buatLangkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks.

10. Di bidang Aplikasi, pilih aplikasi yang Anda unggah ke Amazon S3Langkah 3: Unggah kode aplikasi ke bucket Amazon S3, lalu pilih Langkah berikutnya.
11. Pada halaman Peran AWS Layanan, pilih Buat Peran.

    Jendela baru terbuka dengan halaman konsol IAM yang menjelaskan peran yang akan dibuat untuk Anda. `AWS-CodePipeline-Service` Dari daftar drop-down Nama kebijakan, pilih Buat kebijakan baru. Pastikan dokumen kebijakan memiliki konten berikut. Pilih Edit untuk mengubah dokumen kebijakan, jika diperlukan.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "opsworks:*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

    Setelah Anda selesai membuat perubahan pada dokumen kebijakan, pilih Izinkan. Perubahan Anda akan ditampilkan di konsol IAM.



> **Note**
>
> Jika pembuatan peran gagal, itu mungkin karena Anda sudah memiliki peran IAM bernama AWS- CodePipeline -Service. Jika Anda telah menggunakan peran AWS-CodePipeline -Service sebelum Mei 2016, peran tersebut mungkin tidak memiliki izin untuk menggunakan AWS OpsWorks Stacks sebagai penyedia penerapan. Dalam kasus ini, Anda harus memperbarui pernyataan kebijakan seperti yang ditunjukkan pada langkah ini. Jika Anda melihat pesan kesalahan, kembali ke awal langkah ini, dan pilih Gunakan peran yang ada, bukan Buat peran. Jika Anda menggunakan peran yang ada, peran tersebut harus memiliki kebijakan yang dilampirkan yang menyertakan izin yang ditampilkan dalam langkah ini. Untuk informasi selengkapnya tentang peran layanan dan pernyataan kebijakannya, lihat Mengedit Kebijakan untuk Peran Layanan IAM.

12. Jika proses pembuatan peran berhasil, halaman IAM akan ditutup, dan Anda akan dikembalikan ke halaman Peran AWS Layanan. Pilih Langkah selanjutnya.
13. Pada halaman Tinjau pipeline Anda, verifikasi pilihan yang ditampilkan di halaman, lalu pilih Buat pipeline.

14. Ketika pipeline Anda siap, pipeline akan mulai menemukan kode sumber Anda dan menerapkan aplikasi Anda ke tumpukan Anda secara otomatis. Proses ini dapat memakan waktu beberapa menit.

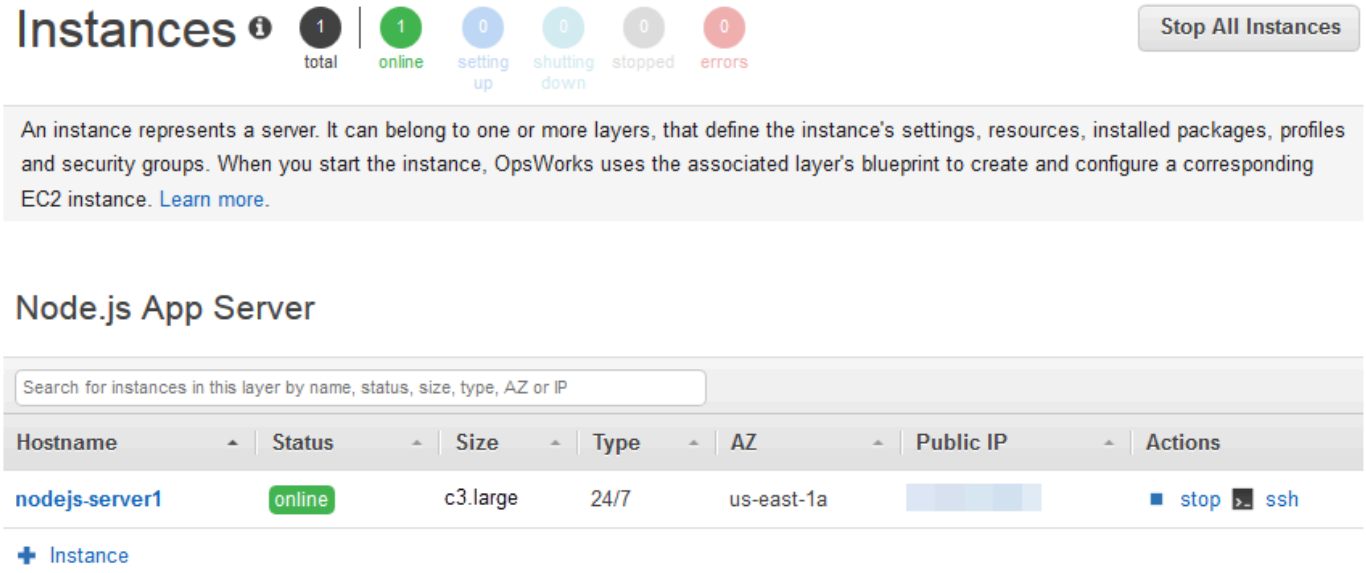
Langkah 6: Memverifikasi penerapan aplikasi di Stacks AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

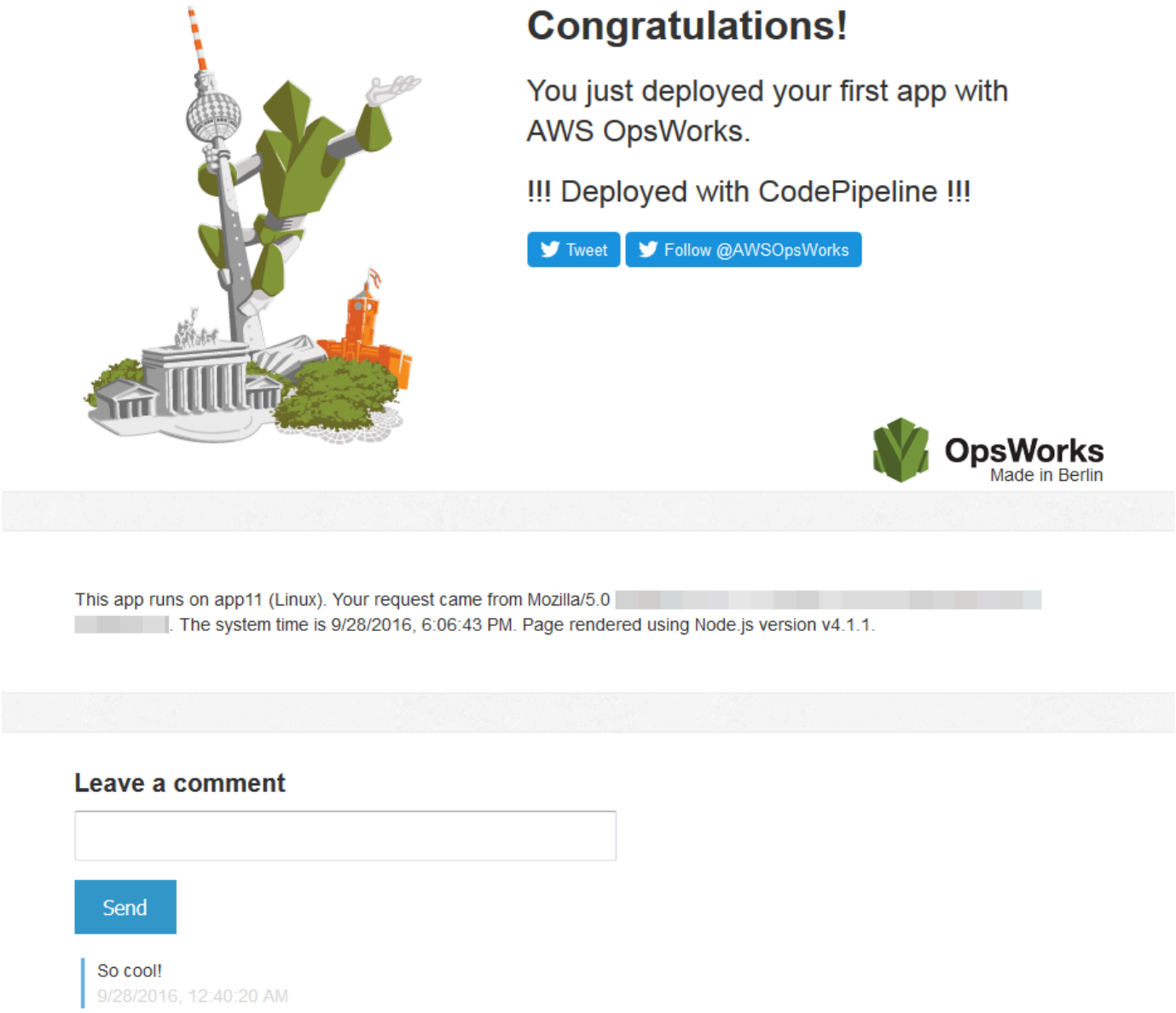
Untuk memverifikasi bahwa aplikasi Node.js CodePipeline telah di-deploy ke stack Anda, masuk ke instance yang Anda buat. Langkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks Anda harus dapat melihat dan menggunakan aplikasi web Node.js.

Untuk memverifikasi penerapan aplikasi di instans AWS OpsWorks Stacks Anda

1. Buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di dasbor AWS OpsWorks Stacks, pilih CodePipelineDemo, lalu pilih Node.js App Server.
3. Di panel navigasi, pilih Instans, lalu pilih alamat IP publik instans yang Anda buat untuk menampilkan aplikasi web.



Aplikasi akan ditampilkan di tab peramban baru.



Langkah 7 (Opsional): Perbarui kode aplikasi untuk melihat CodePipeline penerapan ulang aplikasi Anda secara otomatis

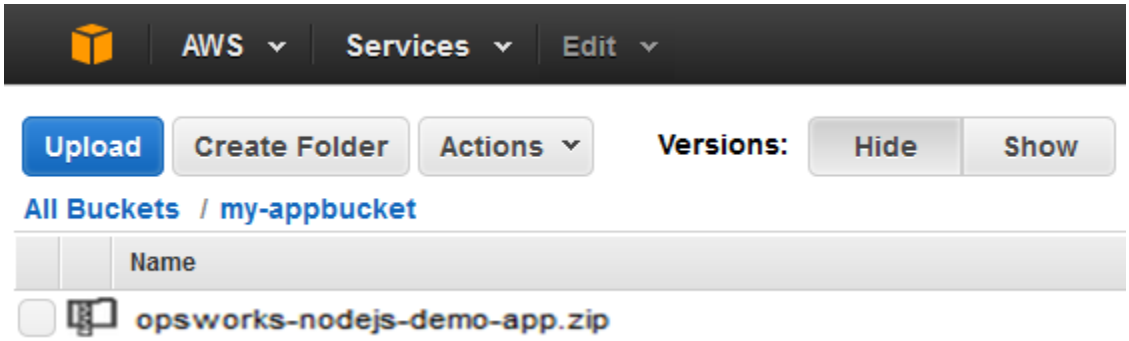
> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

> Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Saat Anda membuat perubahan pada kode di aplikasi atau buku masak yang telah Anda gunakan CodePipeline, artefak yang diperbarui akan diterapkan secara otomatis CodePipeline ke instance target Anda (dalam hal ini, ke tumpukan Stacks target). AWS OpsWorks Bagian ini menunjukkan redeployment otomatis ketika Anda memperbarui kode dalam contoh aplikasi Node.js Anda. Jika Anda masih memiliki kode aplikasi untuk panduan ini disimpan secara lokal, dan tidak ada orang lain yang membuat perubahan pada kode sejak Anda memulai penelusuran, Anda dapat melewati langkah 1-4 dari prosedur ini.

Untuk mengedit kode di aplikasi sampel

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Buka bucket tempat Anda menyimpan contoh aplikasi Node.js.

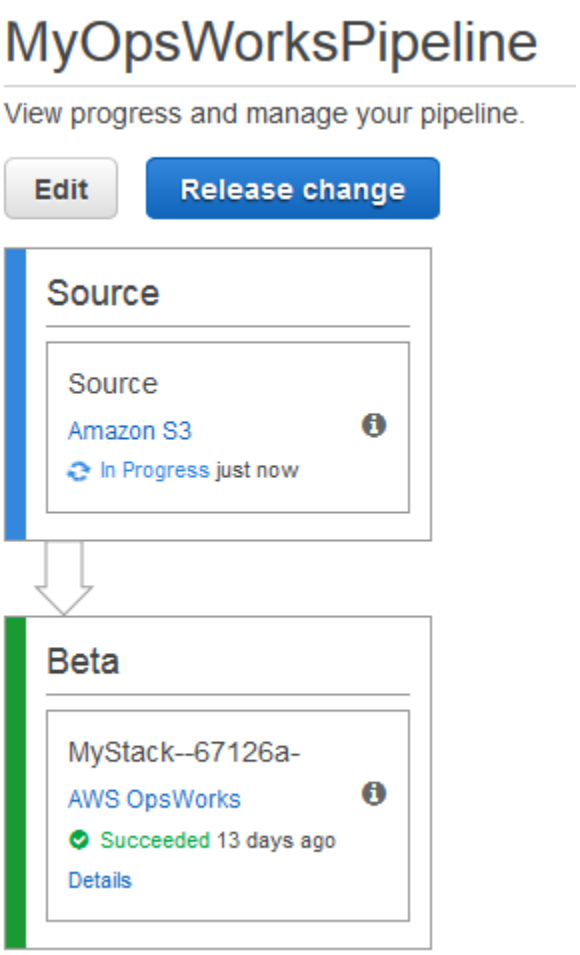


3. Pilih file ZIP yang berisi aplikasi. Di menu Tindakan, pilih Unduh.
4. Di kotak dialog, buka menu konteks (klik kanan), pilih Unduh, lalu simpan file ZIP ke lokasi yang nyaman. Pilih OK.
5. Ekstrak konten file ZIP ke lokasi yang nyaman. Anda mungkin perlu mengubah izin pada folder yang diekstrak dan subfolder serta isinya untuk memungkinkan pengeditan. Di `opsworks-nodejs-demo-app\views` folder, buka `header.html` file untuk diedit.
6. Cari frasa,`You just deployed your first app with`. Ganti kata `deployed` dengan`updated`. Pada baris berikutnya, ubah `AWS OpsWorks.` ke `AWS OpsWorks and AWS CodePipeline.` Jangan mengedit apa pun kecuali teks.

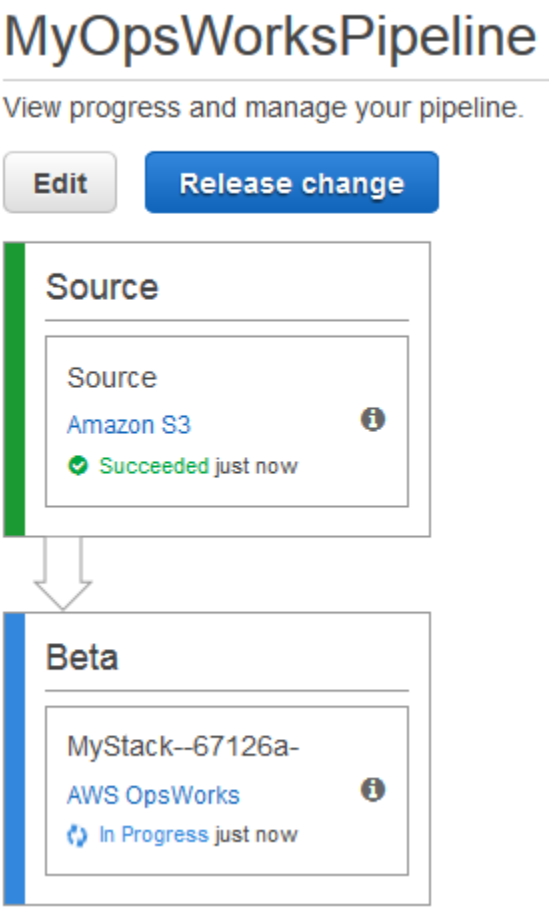
```
<div id="main" role="main">LF
  <div class="container">LF
    <div class="hero-unit">LF
      <div class="robot">LF
        <h1>Congratulations!</h1>LF
        <h2>LF
          You just updated your first app with<br/>LF
          AWS OpsWorks and AWS CodePipeline.LF
        </h2>LF
```

7. Simpan dan tutup file `header.html`.
8. Zip `opsworks-nodejs-demo-app` folder, dan simpan file ZIP ke lokasi yang nyaman. Jangan mengubah nama file ZIP.
9. Unggah file ZIP baru ke bucket Amazon S3 Anda. Dalam panduan ini, nama ember adalah. `my-appbucket`
10. Buka CodePipeline konsol, dan buka pipeline AWS OpsWorks Stacks (MyOpsWorksPipeline). Pilih Perubahan Rilis.

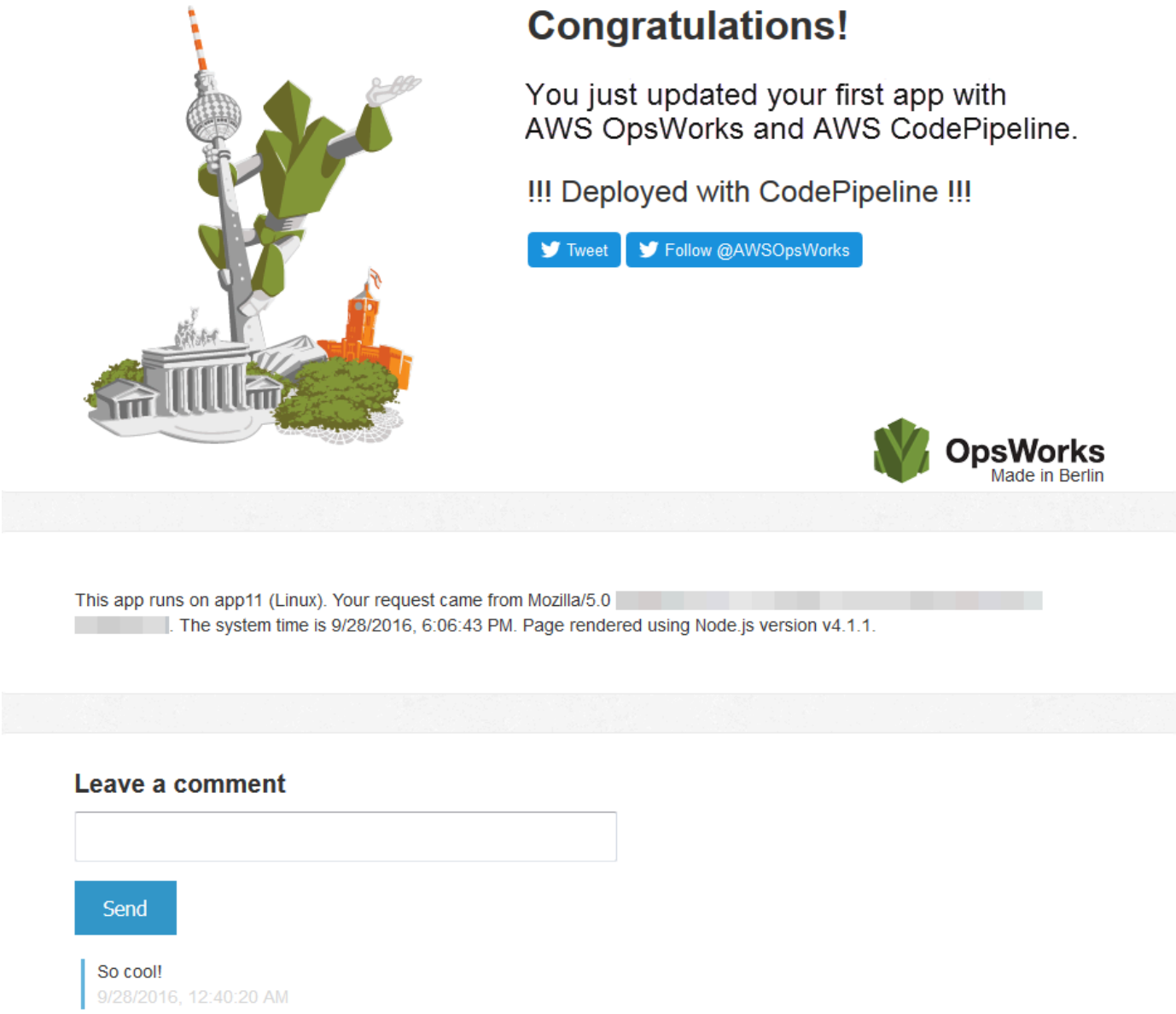
    (Anda dapat menunggu CodePipeline untuk mendeteksi perubahan kode dari versi aplikasi yang diperbarui di bucket Amazon S3 Anda. Untuk menghemat waktu Anda, panduan ini menginstruksikan Anda untuk memilih Release Change.)
11. Amati saat CodePipeline berjalan melalui tahapan pipa. Pertama, CodePipeline mendeteksi perubahan pada artefak sumber.



CodePipeline mendorong kode yang diperbarui ke tumpukan Anda di AWS OpsWorks Stacks.



12. Ketika kedua tahap pipeline telah berhasil diselesaikan, buka tumpukan Anda di AWS OpsWorks Stacks.
13. Pada halaman properti tumpukan, pilih Instans.
14. Di kolom IP Publik, pilih alamat IP publik instans Anda untuk melihat teks aplikasi yang diperbarui.



Langkah 8 (Opsional): Bersihkan Sumber Daya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk membantu mencegah tagihan yang tidak diinginkan ke AWS akun Anda, Anda dapat menghapus AWS sumber daya yang Anda gunakan untuk panduan ini. AWS Sumber daya ini mencakup tumpukan AWS OpsWorks Stacks, peran IAM dan profil instance, dan pipeline yang Anda buat. CodePipeline Namun, Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks dan CodePipeline. Jika Anda ingin menyimpan sumber daya ini, Anda telah menyelesaikan panduan ini.

Untuk menghapus aplikasi dari tumpukan

Karena Anda tidak membuat atau menerapkan aplikasi sebagai bagian dari AWS CloudFormation template Anda, hapus aplikasi pengujian Node.js sebelum Anda menghapus tumpukan AWS CloudFormation.

1. Di konsol AWS OpsWorks Stacks, di panel navigasi layanan, pilih Aplikasi.
2. Di halaman Aplikasi, pilih Aplikasi Demo Node.js, lalu di Tindakan, pilih hapus. Saat Anda diminta untuk mengonfirmasi, pilih Hapus. AWS OpsWorks Stacks akan menghapus aplikasi.

Untuk menghapus tumpukan

Karena Anda membuat tumpukan dengan menjalankan AWS CloudFormation template, Anda dapat menghapus tumpukan, termasuk lapisan, instance, profil instance, dan grup keamanan yang dibuat template, di AWS CloudFormation konsol.

1. Buka AWS CloudFormation konsol.
2. Di dasbor AWS CloudFormation konsol, pilih tumpukan yang Anda buat. Pada menu Actions, pilih Delete Stack. Saat Anda diminta untuk mengonfirmasi, pilih Ya, Hapus.
3. Tunggu DELETE_COMPLETE muncul di kolom Status untuk tumpukan.

Untuk menghapus alur

1. Buka CodePipeline konsol.
2. Di CodePipeline dasbor, pilih pipeline yang Anda buat untuk panduan ini.
3. Pada halaman pipeline, pilih Edit.

4. Pada halaman Edit, pilih Hapus. Ketika diminta untuk mengonfirmasi, pilih Hapus.

AWS CodePipeline dengan AWS OpsWorks Tumpukan - Chef 11 Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS CodePipelinememungkinkan Anda membuat pipeline pengiriman berkelanjutan yang melacak perubahan kode dari sumber seperti CodeCommit, Amazon Simple Storage Service (Amazon S3), atau. GitHub Contoh dalam topik ini menjelaskan cara membuat dan menggunakan pipeline sederhana dari CodePipeline sebagai alat penyebaran untuk kode yang Anda jalankan di lapisan AWS OpsWorks Stacks. Dalam contoh ini, Anda membuat pipeline untuk aplikasi PHP sederhana, lalu menginstruksikan AWS OpsWorks Stacks untuk menjalankan aplikasi pada semua instance dalam lapisan dalam tumpukan Chef 11.10 (dalam hal ini, satu instance).

> **Note**
>
> Topik ini menjelaskan cara menggunakan pipeline untuk menjalankan dan memperbarui aplikasi di tumpukan Chef 11.10. Untuk informasi tentang cara menggunakan pipeline untuk menjalankan dan memperbarui aplikasi di tumpukan Chef 12, lihatAWS CodePipeline dengan AWS OpsWorks Tumpukan - Chef 12 Stacks. Konten yang dikirimkan ke bucket Amazon S3 mungkin berisi konten pelanggan. Untuk informasi selengkapnya tentang menghapus data sensitif, lihat Bagaimana Cara Mengosongkan Bucket S3 Bucket? atau Bagaimana Cara Menghapus bucket S3 Bucket? .

Topik

- Prasyarat
- Skenario lainnya yang didukung
- Langkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks

- Langkah 2: Unggah kode aplikasi ke bucket Amazon S3 Amazon S3
- Langkah 3: Menambahkan aplikasi Anda ke AWS OpsWorks Stacks
- Langkah 4: Buat alur di CodePipeline
- Langkah 5: Memverifikasi penerapan aplikasi di Stacks AWS OpsWorks
- Langkah 6 (Opsional): Perbarui kode aplikasi untuk melihat CodePipeline penerapan ulang aplikasi Anda secara otomatis
- Langkah 7 (Opsional): Bersihkan Sumber Daya

Prasyarat

Sebelum Anda memulai panduan ini, pastikan Anda memiliki izin administrator untuk melakukan semua tugas berikut. Anda dapat menjadi anggota grup yang menerapkan AdministratorAccesskebijakan tersebut, atau Anda dapat menjadi anggota grup yang memiliki izin dan kebijakan yang ditampilkan dalam tabel berikut. Sebagai praktik keamanan terbaik, Anda harus menjadi anggota grup yang memiliki izin untuk melakukan tugas berikut, alih-alih menetapkan izin yang diperlukan untuk pengguna individual.

Untuk informasi selengkapnya tentang membuat grup keamanan di IAM dan menetapkan izin ke grup, lihat Membuat grup pengguna IAM. Untuk informasi selengkapnya tentang mengelola izin AWS OpsWorks Tumpukan, lihat Praktik Terbaik: Mengelola Izin.

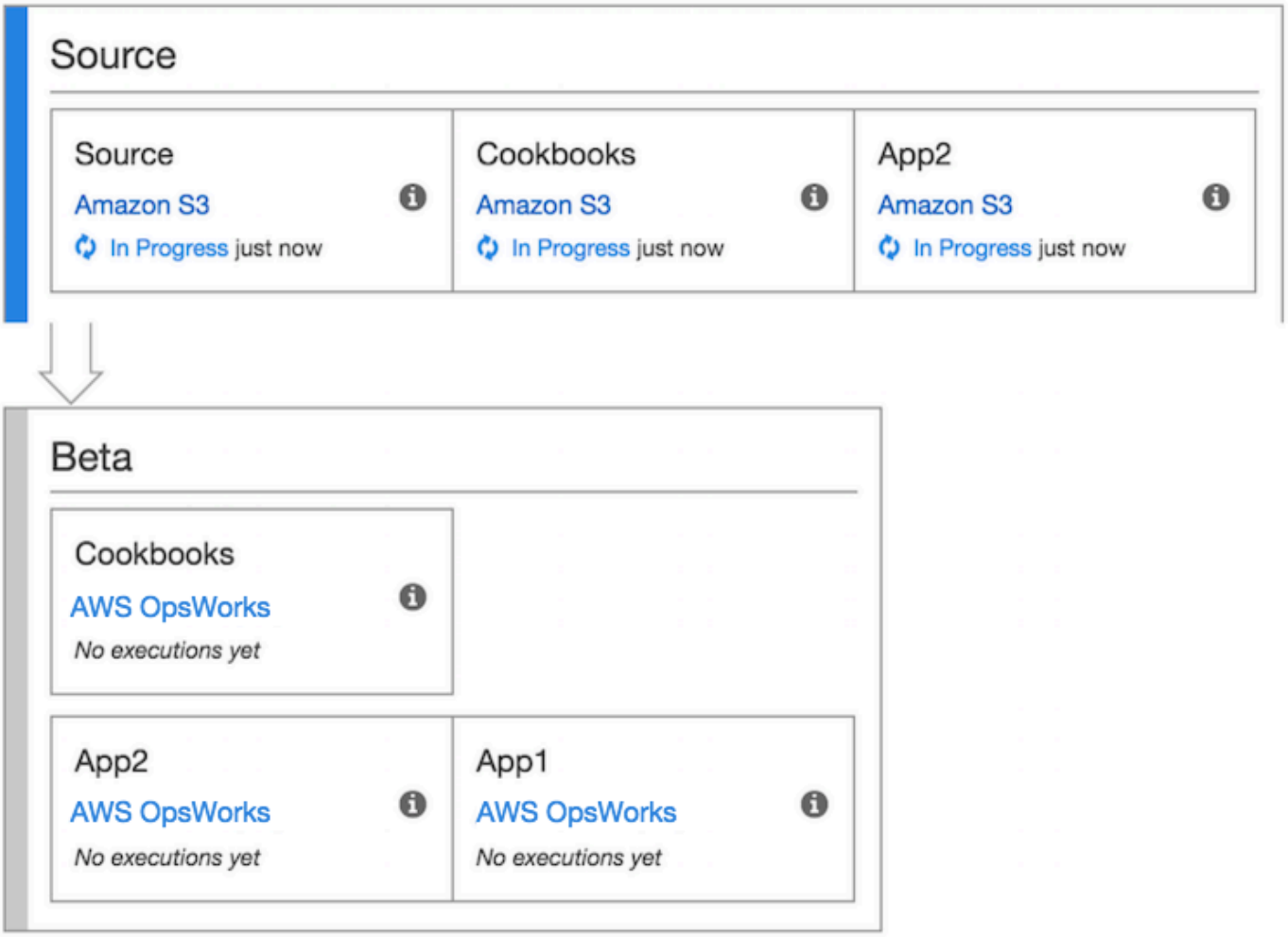
Izin	Kebijakan yang Direkomendasikan untuk Dilampirkan ke Grup
Buat dan edit tumpukan, lapisan, dan instance di AWS OpsWorks Stacks.	AWSOpsWorks_FullAccess
Buat, edit, dan jalankan template di AWS CloudFormation.	AmazonCloudFormationFullAccess
Buat, edit, dan akses bucket Amazon S3.	AmazonS3 FullAccess
Buat, edit, dan jalankan pipeline CodePipeline, terutama pipeline yang menggunakan AWS OpsWorks Stacks sebagai penyedia.	AWSCodePipeline_FullAccess

Anda juga harus memiliki Amazon EC2 key pair. Anda akan diminta untuk memberikan nama key pair ini ketika Anda menjalankan AWS CloudFormation template yang membuat sample stack, layer, dan instance dalam panduan ini. Untuk informasi selengkapnya tentang mendapatkan key pair di EC2 konsol Amazon, lihat Membuat Pasangan Kunci di EC2 dokumentasi Amazon. Pasangan kunci harus berada di Wilayah US East (N. Virginia). Anda dapat menggunakan key pair yang sudah ada jika Anda sudah memilikinya di wilayah tersebut.

Skenario lainnya yang didukung

Panduan ini membuat pipeline sederhana yang mencakup satu tahap Sumber dan satu Penerapan. Namun, Anda dapat membuat pipeline yang lebih kompleks yang menggunakan AWS OpsWorks Stacks sebagai penyedia. Berikut ini adalah contoh pipa dan skenario yang didukung:

- Anda dapat mengedit pipeline untuk menambahkan buku masak Chef ke tahap Sumber dan target terkait untuk buku masak yang diperbarui ke tahap Deploy. Dalam hal ini, Anda menambahkan tindakan Deploy yang memicu pembaruan buku masak Anda saat Anda membuat perubahan pada sumbernya. Buku masak yang diperbarui diterapkan sebelum aplikasi Anda.
- Anda dapat membuat pipeline yang kompleks, dengan buku masak khusus dan beberapa aplikasi, dan menerapkan ke AWS OpsWorks tumpukan Stacks. Pipeline melacak perubahan pada sumber aplikasi dan buku masak, dan digunakan kembali ketika Anda telah membuat perubahan. Berikut ini adalah contoh alur yang serupa dan kompleks:



Untuk informasi lebih lanjut tentang bekerja dengan CodePipeline, lihat CodePipelinedokumentasi.

Langkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk menggunakan AWS OpsWorks Stacks sebagai penyedia penyebaran untuk pipeline, Anda harus terlebih dahulu memiliki tumpukan, lapisan, dan setidaknya satu instance di lapisan. Meskipun Anda dapat membuat tumpukan di AWS OpsWorks Stacks dengan mengikuti petunjuk di Memulai dengan Tumpukan Linux atau Memulai dengan Tumpukan Windows, untuk menghemat waktu Anda, contoh ini menggunakan AWS CloudFormation templat untuk membuat tumpukan, lapisan, dan instance Chef 11.10 berbasis Linux. Instance yang dibuat oleh template ini menjalankan Amazon Linux 2016.03, dan memiliki tipe instance. `c3.large`

> **Important**
>
> AWS CloudFormation Template harus disimpan dan dijalankan di wilayah yang sama dengan bucket Amazon S3 tempat Anda kemudian mengunggah aplikasi dan wilayah yang sama tempat Anda nantinya membuat pipeline. CodePipeline Saat ini, CodePipeline mendukung penyedia AWS OpsWorks Stacks di Wilayah US East (N. Virginia) (us-east-1) saja. Semua sumber daya dalam panduan ini harus dibuat di Wilayah US East (N. Virginia).
> Jika pembuatan tumpukan gagal, Anda mungkin mendekati jumlah maksimum peran IAM yang diizinkan untuk akun Anda. Pembuatan tumpukan juga bisa gagal jika akun Anda tidak dapat meluncurkan `c3.large` instance dengan tipe instans. Misalnya, jika Anda menggunakan Tingkat AWS Gratis, Anda mungkin menerima kesalahan seperti`Root device type: must be included in EBS`. Jika akun Anda memiliki batasan pada jenis instans yang diizinkan untuk dibuat, seperti batasan yang diberlakukan oleh Tingkat AWS Gratis, coba ubah nilai `InstanceType` parameter di blok instans templat menjadi jenis instans yang dapat digunakan akun Anda.

Untuk membuat stack, layer, dan instance menggunakan AWS CloudFormation

1. Salin AWS CloudFormation template berikut ke dalam dokumen teks biasa baru. Simpan file ke lokasi yang nyaman di komputer lokal Anda, dan beri nama NewOpsWorksStack.template, atau nama lain yang nyaman bagi Anda.

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

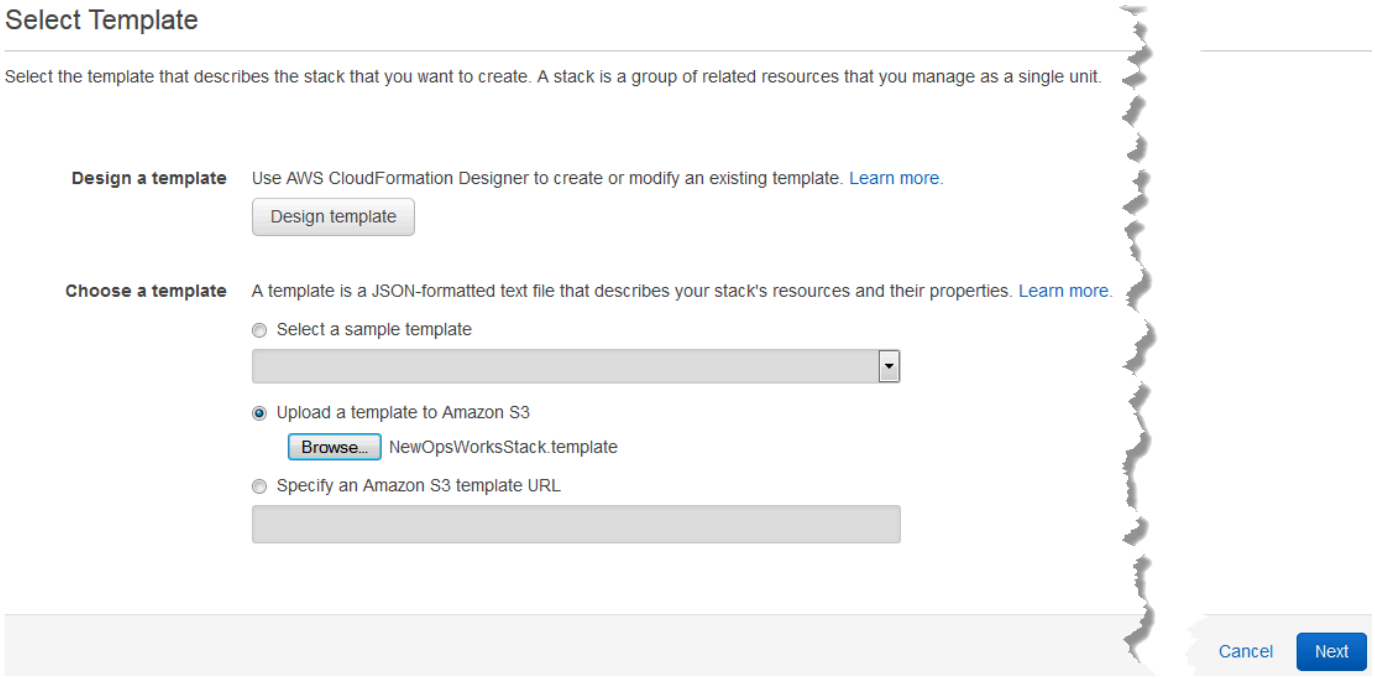
```
    "OpsWorksServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  {
                    "Fn::FindInMap": [
                      "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. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
3. Di AWS CloudFormation halaman beranda, pilih Buat tumpukan.
4. Pada halaman Pilih Templat, di area Pilih templat, pilih Unggah templat ke Amazon S3, lalu pilih Jelajahi.
5. Jelajahi AWS CloudFormation template yang Anda simpan di langkah 1, lalu pilih Buka. Pada halaman Pilih Templat, pilih Selanjutnya.

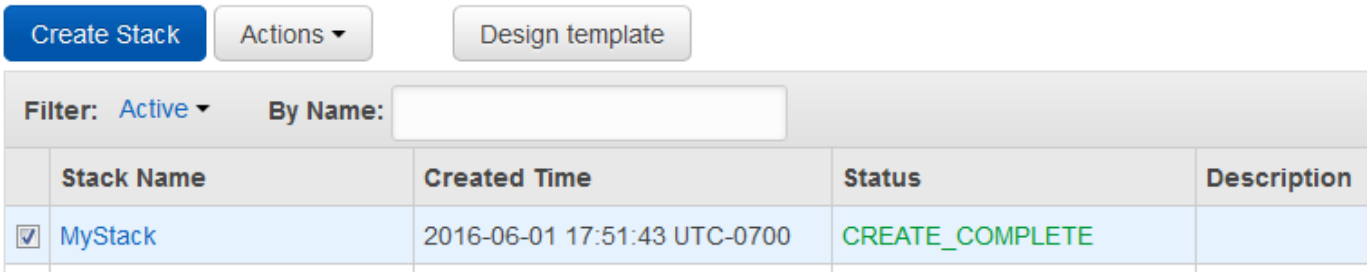


6. Pada halaman Tentukan Detail, beri nama tumpukan MyStack, atau nama tumpukan apa pun yang unik untuk akun Anda. Jika Anda memilih nama yang berbeda untuk tumpukan Anda, ubah nama tumpukan di seluruh panduan ini.
7. Di area Parameter, berikan nama EC2 key pair yang ingin Anda gunakan untuk mengakses instance AWS OpsWorks Stacks Anda setelah dibuat. Pilih Berikutnya.
8. Pada halaman Opsi, pilih Selanjutnya. (Pengaturan di halaman ini tidak diperlukan untuk panduan ini.)
9. AWS CloudFormation Template yang Anda gunakan dalam panduan ini menciptakan peran IAM, profil instance, dan instance.

> **Important**
>
> Sebelum Anda memilih Buat, pilih Biaya untuk memperkirakan biaya yang mungkin dikenakan AWS untuk membuat sumber daya dengan templat ini.

Jika membuat sumber daya IAM dapat diterima, pilih kotak centang Saya mengakui bahwa templat ini dapat CloudFormation menyebabkan AWS membuat sumber daya IAM, lalu pilih Buat. Jika membuat sumber daya IAM tidak dapat diterima, Anda tidak dapat melanjutkan prosedur ini.

10. Di AWS CloudFormation dashboard, Anda dapat melihat kemajuan pembuatan tumpukan. Sebelum Anda melanjutkan ke langkah berikutnya, tunggu hingga CREATE_COMPLETE ditampilkan di kolom Status.



Untuk memverifikasi pembuatan tumpukan di AWS OpsWorks Stacks

1. Buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di dasbor AWS OpsWorks Stacks, lihat tumpukan yang Anda buat.



3. Buka tumpukan, dan lihat layer dan instance. Perhatikan bahwa layer dan instance dibuat dengan nama dan metadata lain yang disediakan dalam template. AWS CloudFormation Anda siap untuk mengunggah aplikasi Anda ke bucket Amazon S3.

Langkah 2: Unggah kode aplikasi ke bucket Amazon S3 Amazon S3

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Karena Anda harus menyediakan tautan ke repositori kode Anda sebagai bagian dari penyiapan pipeline, siapkan repositori kode sebelum membuat pipeline. Di panduan ini, Anda mengunggah aplikasi PHP ke bucket Amazon S3.

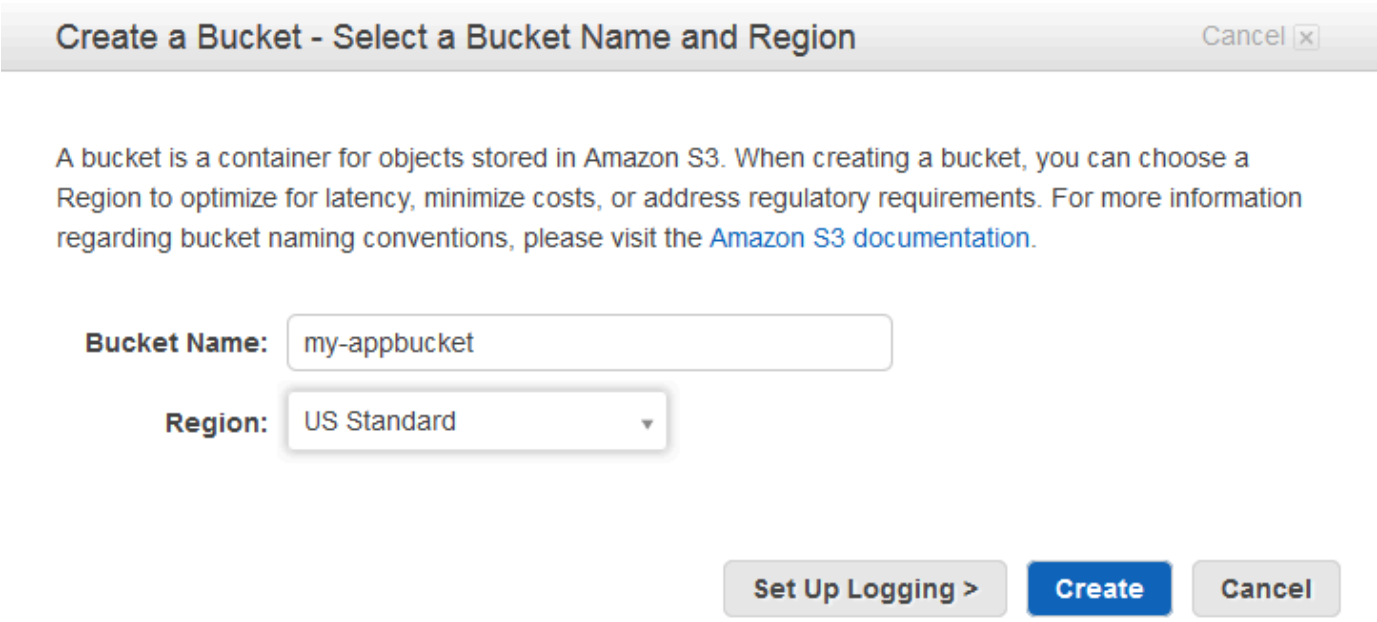
Meskipun CodePipeline dapat menggunakan kode langsung dari GitHub atau CodeCommit sebagai sumber, panduan ini menunjukkan cara menggunakan bucket Amazon S3. Bucket Amazon S3 memungkinkan CodePipeline untuk mendeteksi perubahan pada kode aplikasi dan menerapkan aplikasi yang diubah secara otomatis. Jika mau, Anda dapat menggunakan bucket yang ada. Pastikan bucket memenuhi kriteria CodePipeline seperti yang dijelaskan dalam Simple Pipeline Walkthrough (Amazon S3 Bucket) dalam dokumentasi. CodePipeline

> **Important**
>
> Bucket Amazon S3 harus berada di wilayah yang sama dengan alur Anda nanti. Saat ini, CodePipeline mendukung penyedia AWS OpsWorks Stacks di Wilayah US East (N. Virginia) (us-east-1) saja. Semua sumber daya dalam panduan ini harus dibuat di Wilayah US East (N. Virginia). Bucket juga harus berversi karena CodePipeline memerlukan sumber berversi. Untuk informasi lebih lanjut, lihat Penggunaan Versioning.

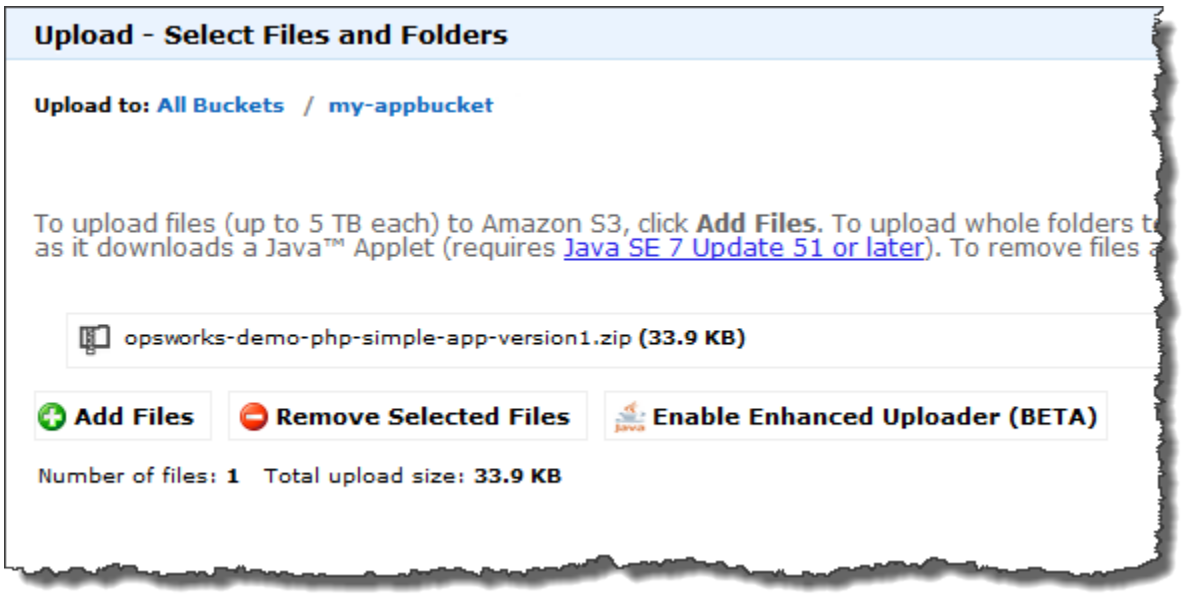
Untuk mengunggah aplikasi Anda ke bucket Amazon S3 Amazon S3

1. Dari GitHub situs web, unduh file ZIP dari aplikasi PHP sampel AWS OpsWorks Stacks, dan simpan ke lokasi yang nyaman di komputer lokal Anda.
2. Pastikan bahwa `index.php` dan `ASSETS` folder berada di tingkat root dari file ZIP yang diunduh. Jika tidak, unzip file, dan buat file ZIP baru yang memiliki file-file ini di tingkat root.
3. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

4. Pilih Buat Bucket.
5. Pada halaman Buat Bucket - Pilih Nama Bucket dan Wilayah, untuk Nama Bucket, ketikkan nama unik untuk bucket Anda. Nama bucket harus unik di semua AWS akun, tidak hanya di akun Anda sendiri. Panduan ini menggunakan nama**`my-appbucket`**, tetapi Anda dapat menggunakannya `my-appbucket-`*`yearmonthday`* untuk membuat nama ember Anda unik. Dari daftar drop-down Region, pilih Standar AS, lalu pilih Buat. Standar AS setara dengan`us-east-1`.



6. Pilih bucket yang Anda buat dari daftar Semua Bucket.
7. Pada halaman bucket, pilih Upload.
8. Pada halaman Unggah - Pilih File dan Folder, pilih Tambahkan file. Jelajahi file ZIP yang Anda simpan di langkah 1, pilih Buka, lalu pilih Mulai Unggah.



9. Setelah unggahan selesai, pilih file ZIP dari daftar file di bucket Anda, lalu pilih Properties.
10. Di panel Properties, salin tautan ke file ZIP Anda, dan catat tautannya. Anda akan memerlukan nama bucket dan bagian nama file ZIP dari tautan ini untuk membuat pipeline Anda.

Langkah 3: Menambahkan aplikasi Anda ke AWS OpsWorks Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sebelum Anda membuat pipeline CodePipeline, tambahkan aplikasi pengujian PHP ke AWS OpsWorks Stacks. Saat membuat pipeline, Anda harus memilih aplikasi yang telah ditambahkan ke AWS OpsWorks Stacks.

Siapkan tautan bucket Amazon S3 dari langkah 10 dari prosedur sebelumnya. Anda akan memerlukan tautan ke bucket tempat Anda menyimpan aplikasi pengujian untuk menyelesaikan prosedur ini.

Untuk menambahkan aplikasi ke AWS OpsWorks Stacks

1. Di konsol AWS OpsWorks Stacks, buka MyStack, dan di panel navigasi, pilih Aplikasi.
2. Pilih Tambahkan aplikasi.
3. Di halaman Tambahkan Aplikasi, berikan informasi berikut ini:

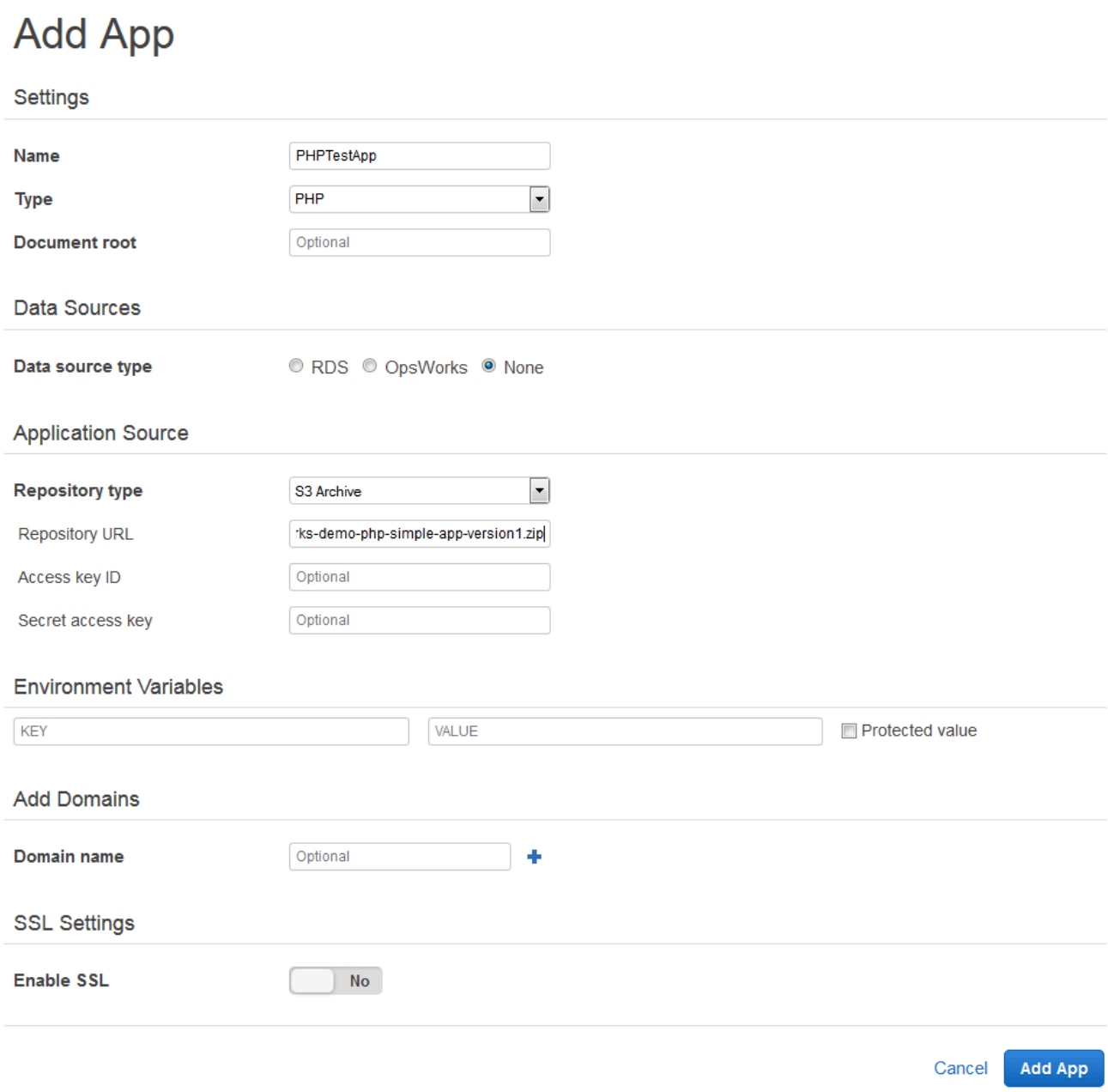
    a. Tentukan nama untuk aplikasi Anda. Panduan ini menggunakan nama. `PHPTestApp`

    b. Di daftar Type menurun, pilih PHP.

    c. Untuk tipe sumber data, pilih Tidak ada. Aplikasi ini tidak memerlukan database eksternal atau sumber data.

    d. Dalam daftar drop-down Jenis repositori, pilih Arsip S3.

    e. Di kotak string URL Repositori, tempel URL yang Anda salin di langkah 10. Langkah 2: Unggah kode aplikasi ke bucket Amazon S3 Amazon S3 Formulir Anda harus serupa dengan berikut ini:



4. Anda tidak perlu mengubah pengaturan lainnya di formulir ini. Pilih Tambahkan Aplikasi.
5. Saat PHPTestaplikasi Aplikasi muncul di daftar di halaman Aplikasi, lanjutkan ke prosedur berikutnyaLangkah 4: Buat alur di CodePipeline.

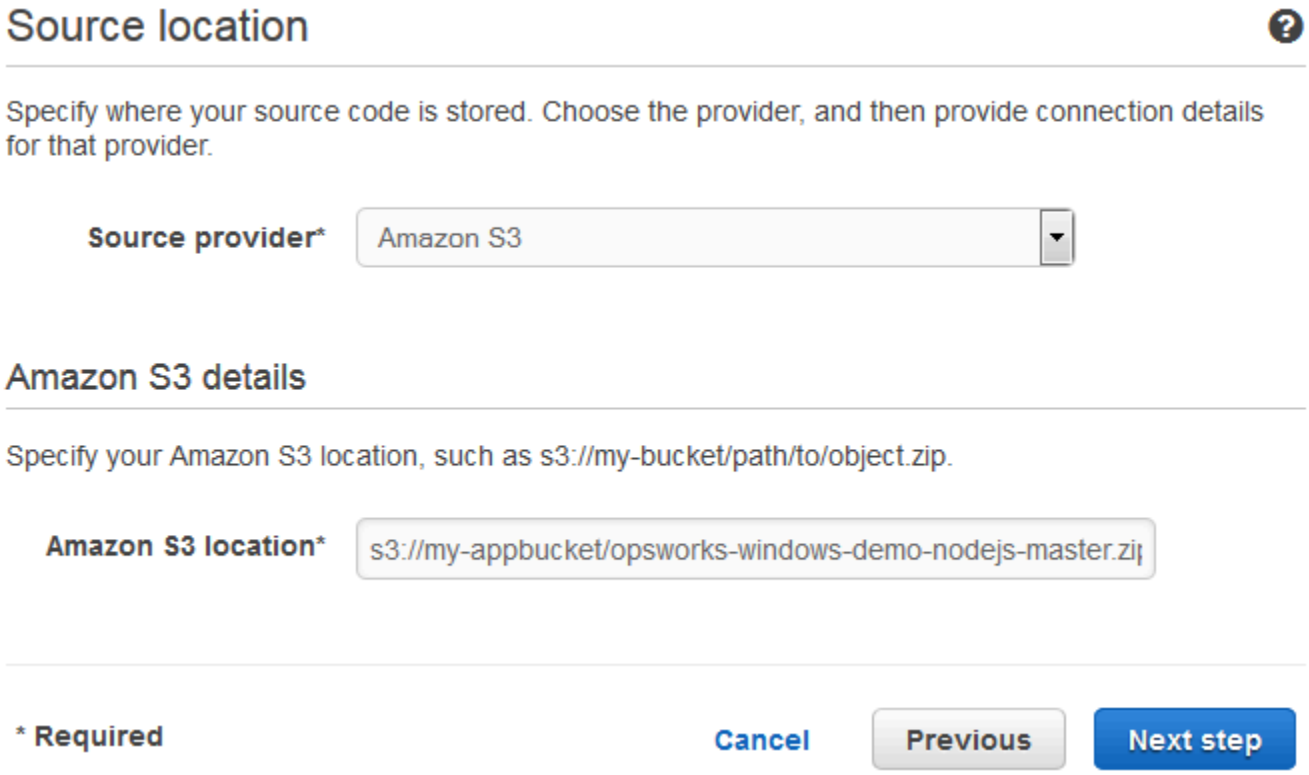
Langkah 4: Buat alur di CodePipeline

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

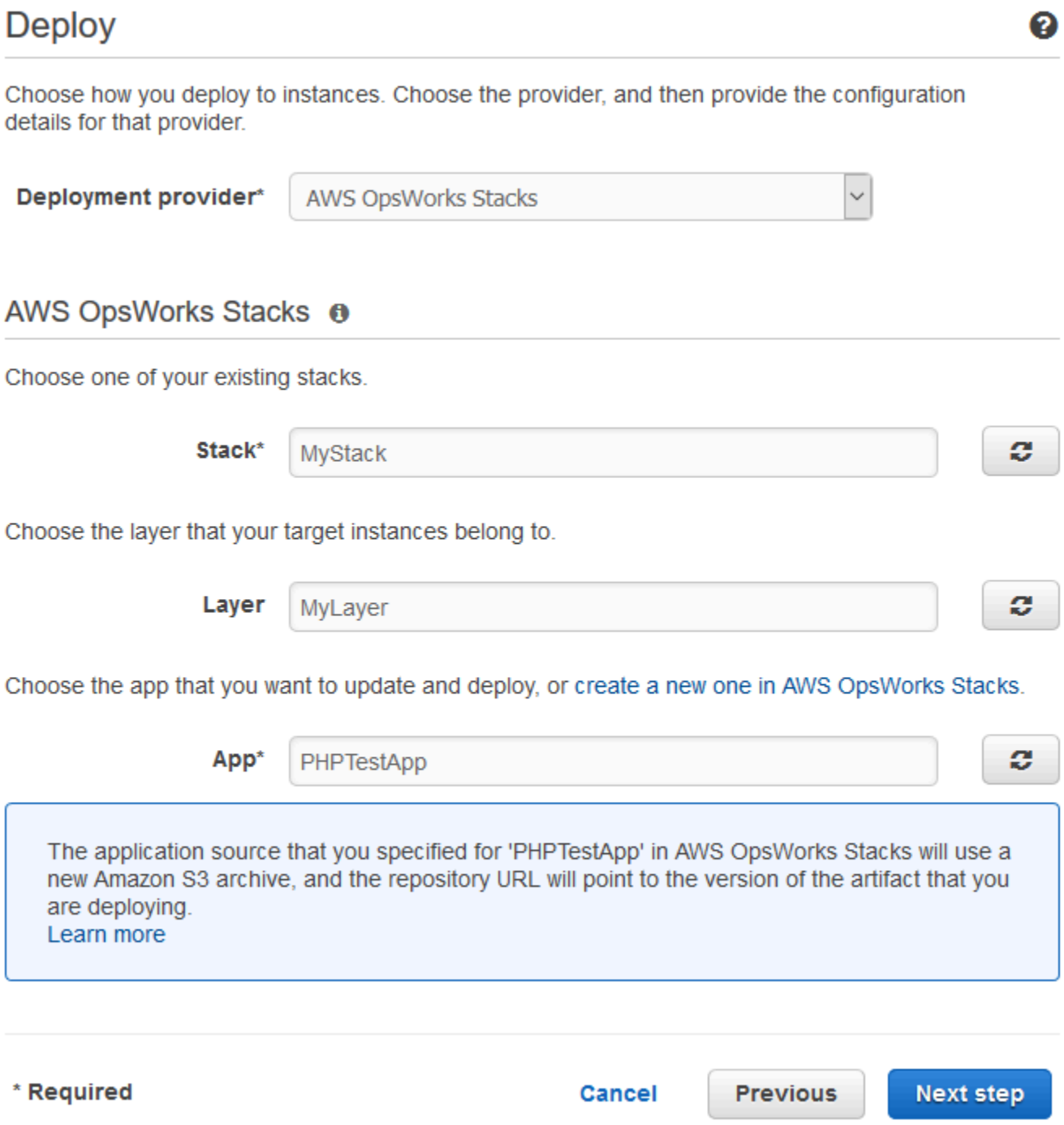
Setelah Anda memiliki tumpukan dengan lapisan dan setidaknya satu instance yang dikonfigurasi di AWS OpsWorks Stacks, buat pipeline CodePipeline dengan AWS OpsWorks Stacks sebagai penyedia untuk menyebarkan aplikasi atau buku masak Chef ke sumber daya Stacks Anda. AWS OpsWorks

Untuk membuat alur

1. Buka CodePipeline konsol di https://console.aws.amazon.com/codepipeline/.
2. Pilih Buat pipeline.
3. Pada CodePipeline halaman Memulai dengan**`MyOpsWorksPipeline`**, ketik, atau nama pipeline lainnya yang unik untuk akun Anda, lalu pilih Langkah berikutnya.
4. Pada halaman Lokasi Sumber, pilih Amazon S3 dari daftar drop-down penyedia Sumber.
5. Di area detail Amazon S3, ketik jalur bucket Amazon S3 Anda, dalam format. **`s3://`*`bucket-name`*`/`*`file name`* Lihat tautan yang Anda catat di langkah 10 dariLangkah 2: Unggah kode aplikasi ke bucket Amazon S3 Amazon S3. Dalam panduan ini, jalannya adalah. `s3://my-appbucket/opsworks-demo-php-simple-app-version1.zip` Pilih Langkah selanjutnya.



6. Pada halaman Build, pilih No Build dari daftar drop-down, lalu pilih Langkah berikutnya.
7. Pada halaman Deploy, pilih AWS OpsWorks Stacks sebagai penyedia penyebaran.



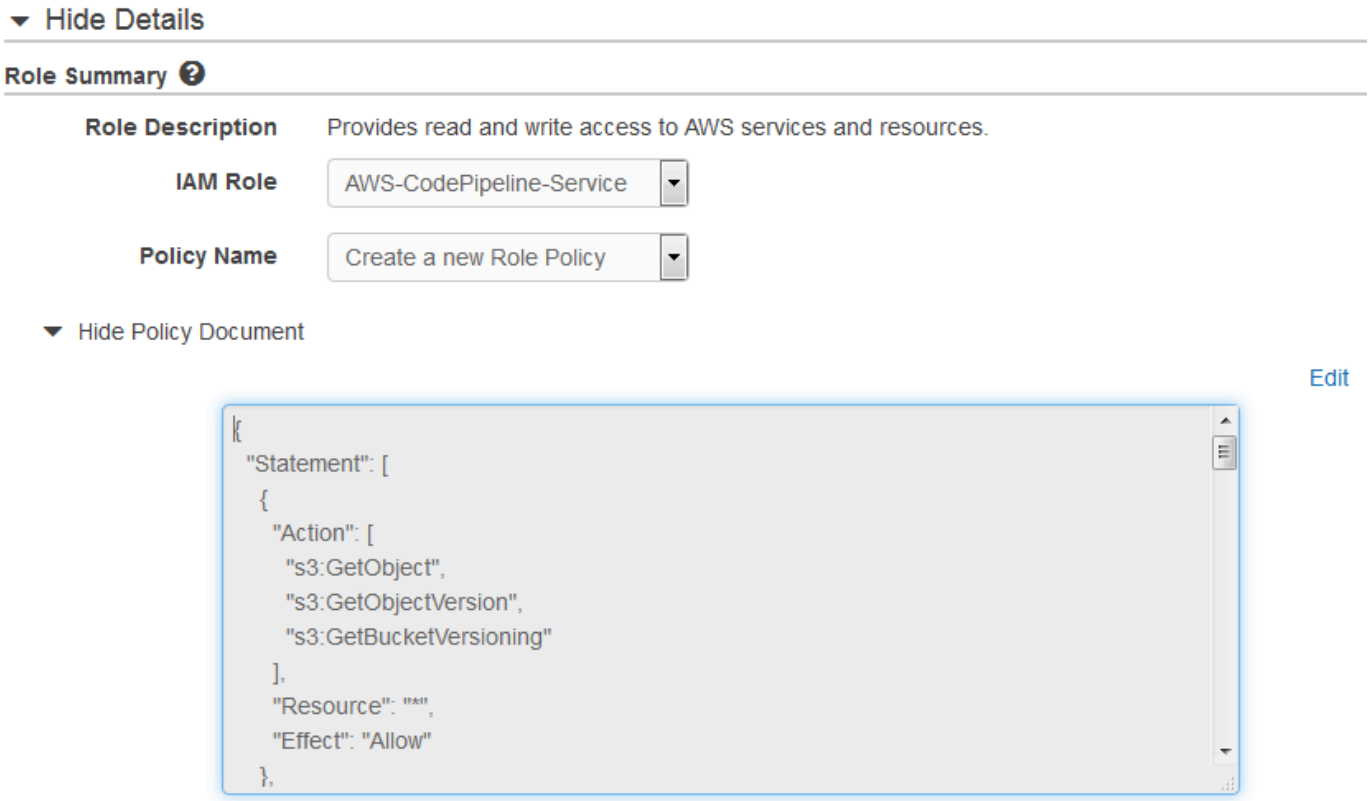
8. Di bidang Stack`MyStack`, ketik, atau nama tumpukan yang Anda buatLangkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks.
9. Di bidang Layer`MyLayer`, ketik, atau nama layer yang Anda buatLangkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks.

10. Di bidang Aplikasi, pilih aplikasi yang Anda unggah ke Amazon S3Langkah 2: Unggah kode aplikasi ke bucket Amazon S3 Amazon S3, lalu pilih Langkah berikutnya.
11. Pada halaman Peran Layanan AWS, pilih Buat Peran.

    Jendela baru terbuka dengan halaman konsol IAM yang menjelaskan peran yang akan dibuat untuk Anda. `AWS-CodePipeline-Service` Dari daftar drop-down Nama kebijakan, pilih Buat kebijakan baru. Pastikan dokumen kebijakan memiliki konten berikut. Pilih Edit untuk mengubah dokumen kebijakan, jika diperlukan.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "opsworks:*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

    Setelah Anda selesai membuat perubahan pada dokumen kebijakan, pilih Izinkan. Perubahan Anda akan dihapus di konsol IAM.



> **Note**
>
> Jika pembuatan peran gagal, itu mungkin karena Anda sudah memiliki peran IAM bernama AWS- CodePipeline -Service. Jika Anda telah menggunakan peran AWS-CodePipeline -Service sebelum Mei 2016, peran tersebut mungkin tidak memiliki izin untuk menggunakan AWS OpsWorks Stacks sebagai penyedia penerapan; dalam hal ini, Anda harus memperbarui pernyataan kebijakan seperti yang ditunjukkan pada langkah ini. Jika Anda melihat pesan kesalahan, kembali ke awal langkah ini, dan pilih Gunakan peran yang ada, bukan Buat peran. Jika Anda menggunakan peran yang ada, peran tersebut harus memiliki kebijakan yang dilampirkan yang menyertakan izin yang ditampilkan dalam langkah ini. Untuk informasi selengkapnya tentang peran layanan dan pernyataan kebijakannya, lihat Mengedit Kebijakan untuk Peran Layanan IAM.

12. Jika proses pembuatan peran berhasil, halaman IAM akan ditutup, dan Anda akan dikembalikan ke halaman Peran Layanan AWS. Pilih Langkah selanjutnya.
13. Pada halaman Tinjau pipeline Anda, verifikasi pilihan yang ditampilkan di halaman, lalu pilih Buat 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. Ketika pipeline Anda siap, pipeline akan mulai menemukan kode sumber Anda dan menerapkan aplikasi Anda ke tumpukan Anda secara otomatis. Proses ini dapat memakan waktu beberapa menit.

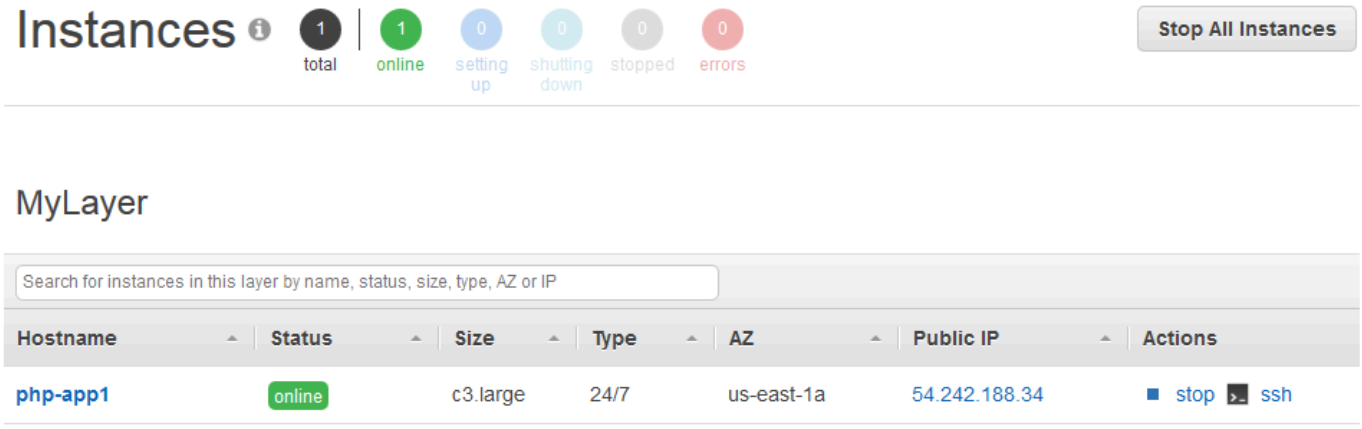
Langkah 5: Memverifikasi penerapan aplikasi di Stacks AWS OpsWorks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

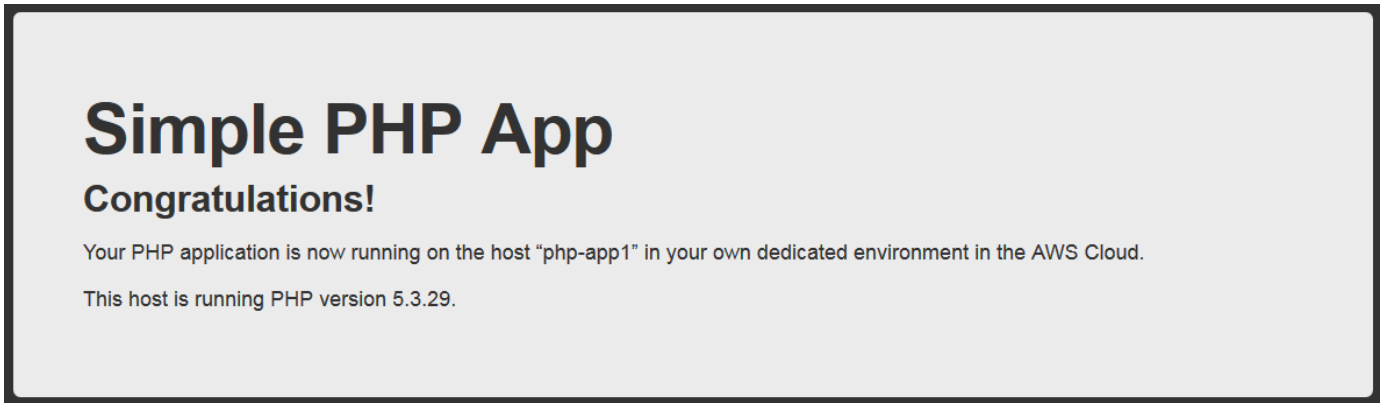
Untuk memverifikasi bahwa aplikasi PHP CodePipeline telah di-deploy ke stack Anda, masuk ke instance yang Anda buat. Langkah 1: Buat tumpukan, lapisan, dan instance di AWS OpsWorks Stacks Anda harus dapat melihat dan menggunakan aplikasi PHP web.

Untuk memverifikasi penerapan aplikasi di instans AWS OpsWorks Stacks Anda

1. Buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di dasbor AWS OpsWorks Stacks, pilih MyStack, lalu pilih MyLayer.
3. Di panel navigasi, pilih Instans, lalu pilih alamat IP publik instance yang Anda buat untuk menampilkan aplikasi web.



Aplikasi ini akan ditampilkan di tab peramban baru.



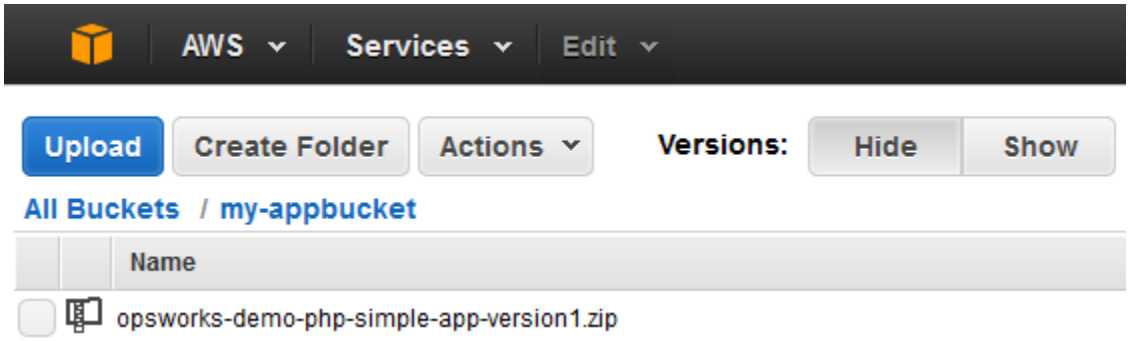
Langkah 6 (Opsional): Perbarui kode aplikasi untuk melihat CodePipeline penerapan ulang aplikasi Anda secara otomatis

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Saat Anda membuat perubahan pada kode di aplikasi atau buku masak yang telah Anda gunakan CodePipeline, artefak yang diperbarui akan diterapkan secara otomatis CodePipeline ke instance target Anda (dalam hal ini, ke tumpukan Stacks target). AWS OpsWorks Bagian ini menunjukkan kepada Anda pemindahan otomatis saat memperbarui kode di aplikasi PHP sampel Anda.

Untuk mengedit kode di aplikasi sampel

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
2. Buka bucket tempat Anda menyimpan contoh aplikasi PHP Anda.



3. Pilih file ZIP yang berisi aplikasi. Di menu Tindakan, pilih Unduh.
4. Di kotak dialog, buka menu konteks (klik kanan), pilih Unduh, lalu simpan file ZIP ke lokasi yang nyaman. Pilih OK.
5. Ekstrak isi file ZIP ke lokasi yang nyaman. Anda mungkin perlu mengubah izin pada folder yang diekstrak dan subfolder serta isinya untuk memungkinkan pengeditan. Di `opsworks-demo-php-simple-app-version1` folder, buka `index.php` file untuk diedit.
6. Cari frasa,`Your PHP application is now running`. Ganti teks, `Your PHP application is now running` dengan`You've just deployed your first app to AWS OpsWorks with AWS CodePipeline,`. Jangan edit variabel.

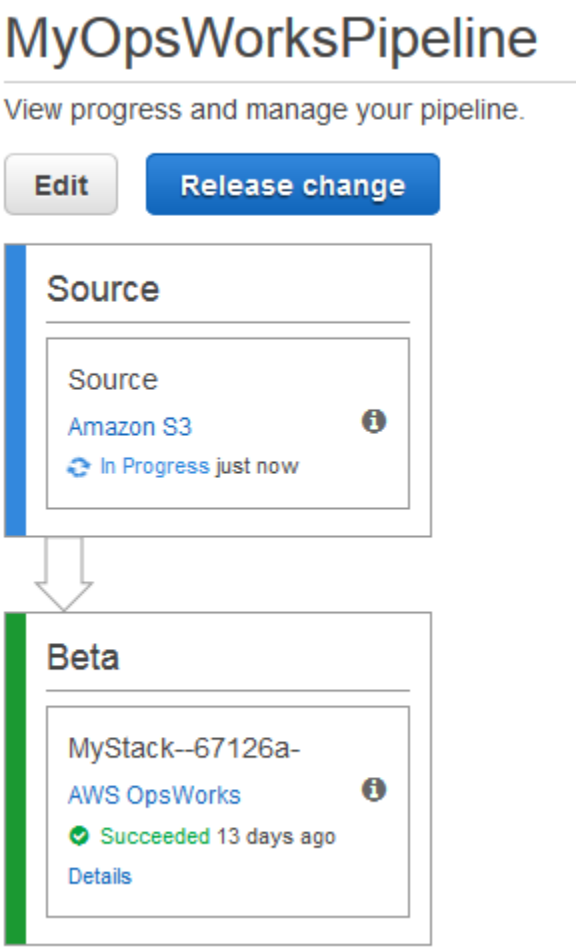
```
<body>
    <div class="container">
        <div class="hero-unit">
            <h1>Simple PHP App</h1>
            <h2>Congratulations!</h2>
            <p>You've just deployed your first app to AWS OpsWorks with AWS CodePipeline,</p>
            <p>on the host &ldquo;<?php echo gethostname(); ?>&rdquo; </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>
```

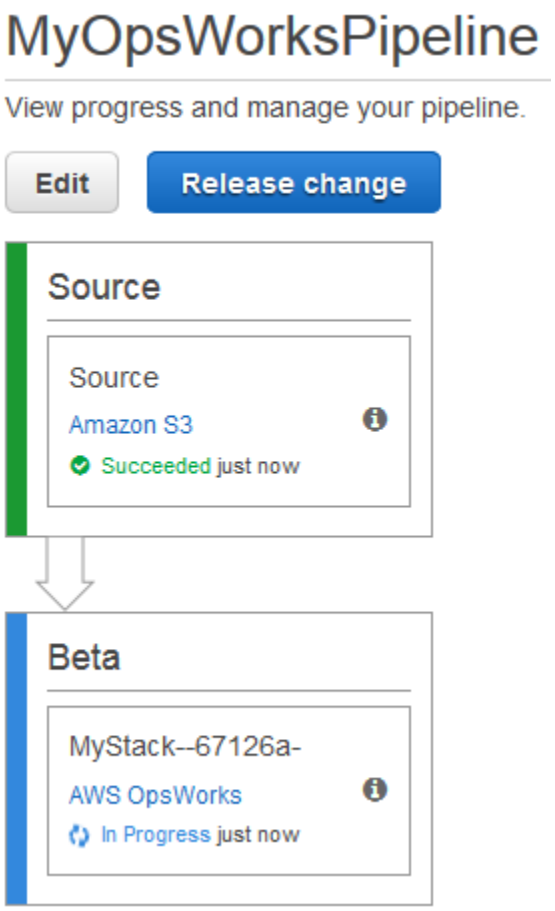
7. Simpan dan tutup file `index.php`.
8. Zip `opsworks-demo-php-simple-app-version1` folder, dan simpan file ZIP ke lokasi yang nyaman. Jangan mengubah nama file ZIP.
9. Unggah file ZIP baru ke bucket Amazon S3 Amazon S3. Dalam panduan ini, nama ember adalah. `my-appbucket`
10. Buka CodePipeline konsol, dan buka pipeline AWS OpsWorks Stacks (MyOpsWorksPipeline). Pilih Perubahan Rilis.

(Anda dapat menunggu CodePipeline untuk mendeteksi perubahan kode dari versi aplikasi yang diperbarui di bucket Amazon S3 Anda. Untuk menghemat waktu, panduan ini menginstruksikan Anda untuk memilih Release Change.)

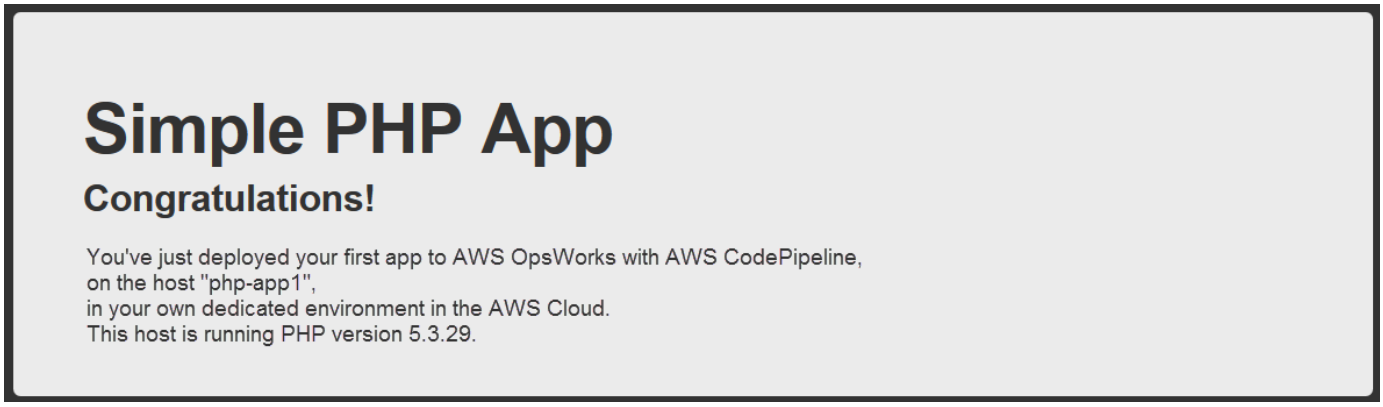
11. Amati saat CodePipeline berjalan melalui tahapan pipa. Pertama, CodePipeline mendeteksi perubahan pada artefak sumber.



CodePipeline mendorong kode yang diperbarui ke tumpukan Anda di AWS OpsWorks Stacks.



12. Ketika kedua tahap pipeline telah berhasil diselesaikan, buka tumpukan Anda di AWS OpsWorks Stacks (MyStack).
13. Pada halaman MyStackproperti, pilih Instans.
14. Di kolom IP Publik, pilih alamat IP publik instans Anda untuk melihat teks aplikasi yang diperbarui.



Langkah 7 (Opsional): Bersihkan Sumber Daya

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk membantu mencegah tagihan yang tidak diinginkan ke akun AWS Anda, Anda dapat menghapus AWS sumber daya yang Anda gunakan untuk panduan ini. AWS Sumber daya ini mencakup tumpukan AWS OpsWorks Stacks, peran IAM dan profil instance, dan pipeline yang Anda buat. CodePipeline Namun, Anda mungkin ingin tetap menggunakan AWS sumber daya ini saat Anda terus mempelajari lebih lanjut tentang AWS OpsWorks Stacks dan CodePipeline. Jika Anda ingin menyimpan sumber daya ini, Anda telah menyelesaikan panduan ini.

Untuk menghapus aplikasi dari tumpukan

Karena Anda tidak membuat atau menerapkan aplikasi sebagai bagian dari AWS CloudFormation template Anda, hapus aplikasi pengujian PHP sebelum Anda menghapus tumpukan AWS CloudFormation.

1. Di konsol AWS OpsWorks Stacks, di panel navigasi layanan, pilih Aplikasi.
2. Di halaman Aplikasi, pilih PHPTestAplikasi, lalu di Tindakan, pilih hapus. Saat Anda diminta untuk mengonfirmasi, pilih Hapus. AWS OpsWorks Stacks akan menghapus aplikasi.

Untuk menghapus tumpukan

Karena Anda membuat tumpukan dengan menjalankan AWS CloudFormation template, Anda dapat menghapus tumpukan, termasuk lapisan, instance, profil instance, dan grup keamanan yang dibuat template, di AWS CloudFormation konsol.

1. Buka AWS CloudFormation konsol.
2. Di dasbor AWS CloudFormation konsol, pilih tumpukan yang Anda buat (MyStack). Pada menu Actions, pilih Delete Stack. Saat Anda diminta untuk mengonfirmasi, pilih Ya, Hapus.
3. Tunggu DELETE_COMPLETE muncul di kolom Status untuk tumpukan.

Untuk menghapus alur

1. Buka CodePipeline konsol.
2. Di CodePipeline dasbor, pilih pipeline yang Anda buat untuk panduan ini.
3. Pada halaman pipeline, pilih Edit.
4. Pada halaman Edit, pilih Hapus. Ketika diminta untuk mengonfirmasi, pilih Hapus.

Menggunakan AWS OpsWorks Stacks CLI

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Antarmuka baris perintah AWS OpsWorks Stacks (CLI) menyediakan fungsionalitas yang sama dengan konsol dan dapat digunakan untuk berbagai tugas. AWS OpsWorks Stacks CLI adalah bagian dari. AWS CLI Untuk informasi selengkapnya, termasuk cara menginstal dan mengonfigurasi AWS CLI, buka What Is the AWS Command Line Interface? . Untuk deskripsi lengkap dari setiap perintah, buka referensi AWS OpsWorks Stacks.

> **Note**
>
> Jika Anda menggunakan workstation berbasis Windows, Anda juga dapat menjalankan AWS Tools for Windows PowerShell untuk melakukan operasi AWS OpsWorks Stacks dari baris perintah. Untuk informasi selengkapnya, lihat AWS Tools untuk Windows PowerShell.

AWS OpsWorks Perintah tumpukan memiliki format umum berikut:

```
aws opsworks --region us-west-1 opsworks command-name [--argument1 value] [...]
```

Jika nilai argumen adalah objek JSON, Anda harus melarikan diri dari " karakter atau perintah mungkin mengembalikan kesalahan bahwa JSON tidak valid. Misalnya, jika objek JSON

adalah`"{"somekey":"somevalue"}"`, Anda harus memformatnya sebagai`"{\"somekey\":\"somevalue\"}"`. Pendekatan alternatif adalah menempatkan objek JSON dalam file dan menggunakannya `file://` untuk memasukkannya ke dalam baris perintah. Contoh berikut membuat aplikasi menggunakan objek sumber aplikasi yang disimpan di 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
```

Sebagian besar perintah mengembalikan satu atau lebih nilai, dikemas sebagai objek JSON. Bagian berikut berisi beberapa contoh. Untuk penjelasan rinci tentang nilai pengembalian untuk setiap perintah, buka referensi AWS OpsWorks Stacks.

> **Note**
>
> AWS CLI perintah harus menentukan wilayah, seperti yang ditunjukkan dalam contoh. Nilai yang valid untuk parameter --region ditunjukkan pada tabel berikut. Untuk menyederhanakan string perintah AWS OpsWorks Stacks Anda, konfigurasikan CLI untuk menentukan wilayah default Anda, sehingga Anda dapat menghilangkan parameter. `--region` Jika Anda biasanya bekerja di beberapa titik akhir regional, jangan mengonfigurasi AWS CLI untuk menggunakan titik akhir regional default. Titik akhir Wilayah Kanada (Tengah) tersedia di API AWS CLI dan hanya; itu tidak tersedia untuk tumpukan yang Anda buat di. AWS Management Console Untuk informasi selengkapnya, lihat Mengonfigurasi Wilayah AWS.
>
> | Nama Wilayah | Kode perintah |
> |---|---|
> | Wilayah AS Timur (Ohio) | us-east-2 |
> | Wilayah US East (N. Virginia) | us-east-1 |
> | Wilayah AS Barat (California Utara) | us-west-1 |
> | Wilayah US West (Oregon) | us-west-2 |
> | Wilayah Kanada (Pusat) | ca-central-1 |
> | Wilayah Eropa (Irlandia) | eu-west-1 |
> | Wilayah Eropa (London) | eu-west-2 |

Nama Wilayah	Kode perintah
Wilayah Eropa (Paris)	eu-west-3
Wilayah Eropa (Frankfurt)	eu-central-1
Wilayah Asia Pasifik (Tokyo)	ap-northeast-1
Wilayah Asia Pacific (Seoul)	ap-northeast-2
Wilayah Asia Pacific (Mumbai)	ap-south-1
Wilayah Asia Pacific (Singapore)	ap-southeast-1
Wilayah Asia Pacific (Sydney)	ap-southeast-2
Wilayah Amerika Selatan (Sao Paulo)	sa-east-1

Untuk menggunakan perintah CLI, Anda harus memiliki izin yang sesuai. Untuk informasi selengkapnya tentang izin AWS OpsWorks Stacks, lihat. Mengelola izin Untuk menentukan izin yang diperlukan untuk perintah tertentu, lihat halaman referensi perintah di referensi AWS OpsWorks Stacks.

Bagian berikut menjelaskan cara menggunakan AWS OpsWorks Stacks CLI untuk melakukan berbagai tugas umum.

Buat sebuah Instance (create-instance)

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah create-instance untuk membuat instance pada tumpukan tertentu.

Topik

- Buat Instance dengan Nama Host Default
- Buat Instance dengan Nama Host Bertema
- Membuat Instance dengan AMI Kustom

Buat Instance dengan Nama Host Default

```
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"
```

Argumennya adalah sebagai berikut:

- `stack-id`— Anda bisa mendapatkan ID tumpukan dari halaman pengaturan tumpukan di konsol (cari OpsWorks ID) atau dengan memanggil deskripsi-tumpukan.
- `layer-ids`— Anda bisa mendapatkan layer IDs dari halaman detail layer di konsol (cari OpsWorks ID) atau dengan memanggil describe-layer. Dalam contoh ini, instance hanya milik satu lapisan.
- `instance-type`— Spesifikasi yang mendefinisikan memori, CPU, kapasitas penyimpanan, dan biaya per jam untuk contoh ini; `m1.large` untuk contoh ini.
- `os`— Sistem operasi instans; Amazon Linux untuk contoh ini.

Perintah mengembalikan objek JSON yang berisi ID contoh, sebagai berikut:

```
{
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Contoh ini menciptakan sebuah instance dengan nama host default, yang hanya integer. Bagian berikut menjelaskan cara membuat instance dengan nama host yang dihasilkan dari tema.

Buat Instance dengan Nama Host Bertema

Anda juga dapat membuat instance dengan nama host bertema. Anda menentukan tema saat Anda membuat tumpukan. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru .Untuk membuat instance, panggil dulu get-hostname-suggestionuntuk menghasilkan nama. Sebagai contoh:

```
C:\>aws opsworks get-hostname-suggestion --region us-west-1 --layer-id 5c8c272a-
f2d5-42e3-8245-5bf3927cb65b
```

Jika Anda menentukan `Layer Dependent` tema default, `get-hostname-suggestion` cukup tambahkan digit ke nama pendek layer. Untuk informasi selengkapnya, lihat Membuat Tumpukan Baru.

Perintah mengembalikan nama host yang dihasilkan.

```
{
    "Hostname": "php-app2",
    "LayerId": "5c8c272a-f2d5-42e3-8245-5bf3927cb65b"
}
```

Anda kemudian dapat menggunakan `hostname` argumen untuk meneruskan nama yang dihasilkan ke`create-instance`, sebagai berikut:

```
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"
```

Membuat Instance dengan AMI Kustom

Perintah create-instance berikut membuat instance dengan AMI kustom, yang harus berasal dari wilayah tumpukan. Untuk informasi selengkapnya tentang cara membuat AMI kustom untuk AWS OpsWorks Tumpukan, lihatMenggunakan 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
```

Argumennya adalah sebagai berikut:

- `stack-id`— Anda bisa mendapatkan ID tumpukan dari halaman pengaturan tumpukan di konsol (cari OpsWorks ID) atau dengan memanggil deskripsi-tumpukan.
- `layer-ids`— Anda bisa mendapatkan layer IDs dari halaman detail layer di konsol (cari OpsWorks ID) atau dengan memanggil describe-layer. Dalam contoh ini, instance hanya milik satu lapisan.
- `instance-type`— Nilai mendefinisikan memori instans, CPU, kapasitas penyimpanan, dan biaya per jam, dan harus kompatibel dengan AMI (`c3.large`untuk contoh ini).
- `os`— Sistem operasi instans, yang harus diatur `Custom` untuk AMI kustom.
- `ami-id`— ID AMI, yang seharusnya terlihat seperti `ami-6c61f104`

> **Note**
>
> Saat Anda menggunakan AMI kustom, pemetaan perangkat blok tidak didukung, dan nilai yang Anda tentukan untuk `--block-device-mappings` opsi akan diabaikan.

Perintah mengembalikan objek JSON yang berisi ID contoh, sebagai berikut:

```
{
    "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Menerapkan Aplikasi (buat-penyebaran)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah create-deployment untuk menerapkan aplikasi ke tumpukan tertentu.

Topik

- Menerapkan Aplikasi

Menerapkan Aplikasi

```
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\"}"
```

Argumennya adalah sebagai berikut:

- `stack-id`— Anda bisa mendapatkan ID tumpukan dari halaman pengaturan tumpukan di konsol (cari OpsWorks ID) atau dengan menelepon`describe-stacks`.
- `app-id`— Anda bisa mendapatkan ID aplikasi dari halaman detail aplikasi (cari OpsWorks ID) atau dengan memanggil deskripsi-aplikasi.
- `command`— Argumen mengambil objek JSON yang menetapkan nama perintah`deploy`, yang menyebarkan aplikasi yang ditentukan ke tumpukan.

  Perhatikan bahwa " karakter dalam objek JSON semuanya lolos. Jika tidak, perintah mungkin mengembalikan kesalahan bahwa JSON tidak valid.

Perintah mengembalikan objek JSON yang berisi ID penyebaran, sebagai berikut:

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

> **Note**
>
> Contoh sebelumnya diterapkan ke setiap instance di tumpukan. Untuk menyebarkan ke subset instance tertentu, tambahkan `instance-ids` argumen dan daftar instance. IDs

Daftar Aplikasi Stack (deskripsikan aplikasi)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah describe-apps untuk membuat daftar aplikasi tumpukan atau mendapatkan detail tentang aplikasi tertentu.

```
aws opsworks --region us-west-1 describe-apps --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Contoh sebelumnya menampilkan objek JSON yang berisi informasi tentang setiap aplikasi. Contoh ini hanya memiliki satu aplikasi. Untuk deskripsi setiap parameter, lihat deskripsi-aplikasi.


```
{
  "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"
```

```
    }
  ]
}
```

Daftar Perintah Stack (deskripsikan perintah)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah deskripsi-perintah untuk membuat daftar perintah tumpukan atau mendapatkan rincian tentang perintah tertentu. Contoh berikut mendapatkan informasi tentang perintah yang telah dieksekusi pada instance tertentu.

```
aws opsworks --region us-west-1 describe-commands --instance-id
 8c2673b9-3fe5-420d-9cfa-78d875ee7687
```

Perintah mengembalikan objek JSON yang berisi rincian tentang setiap perintah. TypeParameter mengidentifikasi nama perintah, menyebarkan atau membatalkan penerapan untuk contoh ini. Untuk deskripsi parameter lainnya, lihat deskripsi-perintah.

```
{
  "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
    }
  ]
}
```

Daftar Deployment Stack (deskripsikan penerapan)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah deskripsi-deployments untuk membuat daftar penerapan tumpukan atau mendapatkan detail tentang penerapan yang ditentukan.

```
aws opsworks --region us-west-1 describe-deployments --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Perintah sebelumnya mengembalikan objek JSON yang berisi rincian tentang setiap penyebaran untuk tumpukan tertentu. Untuk deskripsi setiap parameter, lihat deskripsi-penerapan.

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

Daftar Alamat IP Elastis Stack (describe-elastic-ips)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan describe-elastic-ipsperintah untuk mencantumkan alamat IP Elastis yang telah terdaftar dengan tumpukan atau dapatkan detail tentang alamat IP Elastis yang ditentukan.

```
aws opsworks --region us-west-2 describe-elastic-ips --instance-id b62f3e04-e9eb-436c-a91f-d9e9a396b7b0
```

Perintah sebelumnya mengembalikan objek JSON yang berisi rincian tentang setiap alamat IP Elastis (satu dalam contoh ini) untuk contoh tertentu. Untuk deskripsi setiap parameter, lihat describe-elastic-ips.

```
{
  "ElasticIps": [
      {
          "Ip": "192.0.2.0",
          "Domain": "standard",
          "Region": "us-west-2"
      }
  ]
}
```

Daftar Instans Stack (deskripsikan contoh)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

> Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah describe-instance untuk membuat daftar instance tumpukan atau mendapatkan detail tentang instance tertentu.

```
C:\>aws opsworks --region us-west-2 describe-instances --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Perintah sebelumnya mengembalikan objek JSON yang berisi rincian tentang setiap instance dalam tumpukan tertentu. Untuk deskripsi setiap parameter, lihat mendeskripsikan contoh.

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

Daftar Tumpukan Akun (deskripsikan tumpukan)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah deskripsikan tumpukan untuk membuat daftar tumpukan akun atau mendapatkan detail tentang tumpukan tertentu.

```
aws opsworks --region us-west-2 describe-stacks
```

Perintah sebelumnya mengembalikan objek JSON yang berisi rincian tentang setiap tumpukan di akun, dua dalam contoh ini. Untuk deskripsi setiap parameter, lihat deskripsikan tumpukan.

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

Daftar Layers Stack (deskripsikan lapisan)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah deskripsi-layer untuk membuat daftar lapisan tumpukan atau mendapatkan detail tentang lapisan tertentu.

```
aws opsworks --region us-west-2 describe-layers --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

Perintah sebelumnya mengembalikan objek JSON yang berisi rincian tentang setiap lapisan dalam tumpukan tertentu — dalam contoh ini, lapisan MySQL dan lapisan kustom. Untuk deskripsi setiap parameter, lihat mendeskripsikan lapisan.

```
{
  "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": []
    }
  ]
}
```

Jalankan Resep (create-deployment)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah create-deployment untuk menjalankan perintah stack dan perintah deployment. Contoh berikut mengeksekusi perintah stack untuk menjalankan resep kustom pada tumpukan tertentu.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
    --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":[\"phpapp::appsetup
\"]}}"
```

`command`Argumen mengambil objek JSON yang diformat sebagai berikut:

- `Name`- Menentukan nama perintah. `execute_recipes`Perintah yang digunakan untuk contoh ini mengeksekusi resep tertentu pada instance tumpukan.
- `Args`- Menentukan daftar argumen dan nilai-nilai mereka. Contoh ini memiliki satu argumen,`recipes`, yang diatur ke resep yang akan dieksekusi,`phpapp::appsetup`.

Perhatikan bahwa " karakter dalam objek JSON semuanya lolos. Jika tidak, perintah mungkin mengembalikan kesalahan bahwa JSON tidak valid.

Perintah mengembalikan ID penyebaran, yang dapat Anda gunakan untuk mengidentifikasi perintah untuk perintah CLI lainnya seperti. `describe-commands`

```
{
    "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

Instal Dependensi (buat-penyebaran)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah create-deployment untuk menjalankan perintah stack dan perintah deployment. Contoh berikut menjalankan perintah `update_dependencies` stack untuk memperbarui dependensi pada instance stack.

```
aws opsworks --region us-west-1 create-deployment --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb
--command "{\"Name\":\"install_dependencies\"}"
```

`command`Argumen mengambil objek JSON dengan `Name` parameter yang nilainya menentukan nama perintah, `install_dependencies` untuk contoh ini. Perhatikan bahwa " karakter dalam objek JSON semuanya lolos. Jika tidak, perintah mungkin mengembalikan kesalahan bahwa JSON tidak valid.

Perintah mengembalikan ID penyebaran, yang dapat Anda gunakan untuk mengidentifikasi perintah untuk perintah CLI lainnya seperti. `describe-commands`

```
{
```

```
    "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

Perbarui Konfigurasi Stack (update-stack)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Gunakan perintah update-stack untuk memperbarui konfigurasi tumpukan tertentu. Contoh berikut memperbarui tumpukan untuk menambahkan JSON kustom ke atribut konfigurasi tumpukan.

```
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
```

Perhatikan bahwa " karakter dalam objek JSON semuanya lolos. Jika tidak, perintah mungkin mengembalikan kesalahan bahwa JSON tidak valid.

> **Note**
>
> Contoh ini juga menentukan peran layanan untuk tumpukan. Anda harus mengatur `service-role-arn` ke ARN peran layanan yang valid atau tindakan akan gagal; tidak ada nilai default. Anda dapat menentukan ARN peran layanan stack saat ini, jika Anda mau, tetapi Anda harus melakukannya secara eksplisit.

`update-stack`Perintah tidak mengembalikan nilai.

Panduan Debugging dan Pemecahan Masalah

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika Anda perlu men-debug resep atau memecahkan masalah layanan, pendekatan terbaik umumnya adalah bekerja melalui langkah-langkah berikut, secara berurutan:

1. Masalah Debugging dan Pemecahan Masalah UmumPeriksa masalah spesifik Anda.
2. Cari Forum AWS OpsWorks Stacks untuk melihat apakah masalah telah dibahas di sana.

   Forum ini mencakup banyak pengguna berpengalaman dan dipantau oleh tim AWS OpsWorks Stacks.
3. Untuk masalah dengan resep, lihatResep Debugging.
4. Hubungi dukungan AWS OpsWorks Stacks atau posting masalah Anda di AWS OpsWorks Stacks Forum.

Bagian berikut memberikan panduan untuk debugging resep. Bagian terakhir menjelaskan masalah debugging dan pemecahan masalah umum dan solusinya.

> **Note**
>
> Setiap Chef run menghasilkan log, yang memberikan deskripsi rinci tentang proses dan merupakan sumber pemecahan masalah yang berharga. Untuk menentukan jumlah detail dalam log, tambahkan `Chef::Log.level`pernyataan ke resep kustom yang menentukan tingkat log yang diinginkan. Nilai default-nya adalah `:info`. Contoh berikut menunjukkan cara mengatur level log Chef ke`:debug`, yang memberikan deskripsi paling rinci tentang proses.
>
> ```
> Chef::Log.level = :debug
> ```

Untuk informasi selengkapnya tentang melihat dan menafsirkan log Chef, lihatLog Koki.

Topik

- Resep Debugging
- Masalah Debugging dan Pemecahan Masalah Umum

Resep Debugging

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Ketika peristiwa siklus hidup terjadi, atau Anda menjalankan perintah tumpukan Execute Recipes, AWS OpsWorks Stacks mengeluarkan perintah ke agen untuk memulai menjalankan Chef Solo pada instance yang ditentukan untuk menjalankan resep yang sesuai, termasuk resep kustom Anda. Bagian ini menjelaskan beberapa cara agar Anda dapat men-debug resep yang gagal.

Topik

- Masuk ke Instance Gagal
- Log Koki
- Menggunakan AWS OpsWorks Stacks Agent CLI

Masuk ke Instance Gagal

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Jika resep gagal, instance akan berakhir di `setup_failed` negara bagian alih-alih online. Meskipun instance tidak online sejauh menyangkut AWS OpsWorks Stacks, EC2 instance sedang berjalan dan seringkali berguna untuk masuk untuk memecahkan masalah. Misalnya, Anda dapat memeriksa apakah aplikasi atau buku masak khusus diinstal dengan benar. Dukungan bawaan AWS OpsWorks Stacks untuk login SSH dan RDP hanya tersedia untuk instance dalam status online. Namun, jika Anda telah menetapkan key pair SSH ke instance, Anda masih dapat masuk, sebagai berikut:

- Instance Linux — Gunakan kunci pribadi pasangan kunci SSH untuk masuk dengan klien SSH pihak ketiga, seperti OpenSSH atau PutTY.

  Anda dapat menggunakan EC2 key pair atau key pair SSH pribadi Anda untuk tujuan ini.
- Instans Windows — Gunakan EC2 kunci pribadi pasangan kunci untuk mengambil kata sandi Administrator instans.

  Gunakan kata sandi itu untuk masuk dengan Klien RDP pilihan Anda. Untuk informasi selengkapnya, lihat Masuk Sebagai Administrator. Anda tidak dapat menggunakan key pair SSH pribadi untuk mengambil kata sandi Administrator.

Log Koki

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Log koki adalah salah satu sumber pemecahan masalah utama Anda, terutama untuk men-debug resep. AWS OpsWorks Stacks menangkap log Chef untuk setiap perintah, dan mempertahankan log untuk 30 perintah terbaru instans. Karena run dalam mode debug, log berisi deskripsi rinci tentang

Chef run, termasuk teks yang dikirim ke `stdout` dan`stderror`. Jika resep gagal, log menyertakan jejak tumpukan Chef.

AWS OpsWorks Tumpukan memberi Anda beberapa cara untuk melihat log Chef. Setelah Anda memiliki informasi log, Anda dapat menggunakannya untuk men-debug resep yang gagal.

> **Note**
>
> Anda juga dapat melihat ekor log tertentu dengan menggunakan SSH untuk terhubung ke instance dan menjalankan perintah `show_log` CLI agen. Untuk informasi selengkapnya, lihat Menampilkan Log Chef.

Topik

- Melihat Log Chef dengan Konsol
- Melihat Log Chef dengan CLI atau API
- Melihat Log Chef pada Instance
- Menafsirkan Log Chef
- Kesalahan Log Chef Umum

Melihat Log Chef dengan Konsol

Cara termudah untuk melihat log Chef adalah pergi ke halaman detail instance. Bagian Log mencakup entri untuk setiap acara dan perintah Execute Recipes. Berikut ini menunjukkan bagian Log instance, dengan perintah konfigurasi dan penyiapan, yang sesuai dengan peristiwa siklus hidup Konfigurasi dan Pengaturan.



Klik tampilkan di kolom Log perintah yang sesuai untuk melihat log Chef yang sesuai. Jika terjadi kesalahan, AWS OpsWorks Stacks secara otomatis membuka log ke kesalahan, yang biasanya di akhir file.

Melihat Log Chef dengan CLI atau API

Anda dapat menggunakan perintah AWS OpsWorks Stacks `describe-commands`CLI atau `DescribeCommands`tindakan API untuk melihat log, yang disimpan di bucket Amazon S3. Berikut ini menunjukkan bagaimana menggunakan CLI untuk melihat salah satu set file log saat ini untuk contoh tertentu. Prosedur untuk menggunakan pada `DescribeCommands` dasarnya serupa.

Untuk menggunakan AWS OpsWorks Stacks untuk melihat log Chef instance

1. Buka halaman detail instance dan salin nilai OpsWorksID-nya.
2. Gunakan nilai ID untuk menjalankan perintah `describe-commands` CLI, sebagai berikut:

```
aws opsworks describe-commands --instance-id 67bf0da2-29ed-4217-990c-d895d51812b9
```

Perintah mengembalikan objek JSON dengan objek tertanam untuk setiap perintah yang telah dieksekusi AWS OpsWorks Stacks pada instance, dengan yang terbaru pertama. `Type`Parameter berisi jenis perintah untuk setiap objek tertanam, `configure` perintah, dan `setup` perintah dalam contoh ini.

```
{
    "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. Salin `LogUrl` nilai ke browser Anda untuk melihat log.

Jika instance memiliki lebih dari beberapa perintah, Anda dapat menambahkan parameter `describe-commands` untuk memfilter perintah mana yang disertakan dalam objek respons. Untuk informasi selengkapnya, lihat mendeskripsikan perintah.

Melihat Log Chef pada Instance

> **Note**
>
> Topik di bagian ini berlaku untuk Chef 12. Untuk informasi tentang lokasi log Chef untuk Chef 11.10 dan rilis yang lebih lama, lihat Pemecahan Masalah Chef 11.10 dan Versi Sebelumnya untuk Linux.

Instans Linux

AWS OpsWorks Stacks menyimpan log Chef setiap instance di `/var/chef/runs` direktorinya. (Untuk contoh Linux, direktori ini juga menyertakan kantong data terkait, disimpan sebagai file berformat JSON.) Anda memerlukan hak istimewa sudo untuk mengakses direktori ini. Log untuk setiap proses berada dalam file bernama `chef.log` di dalam subdirektori run individual.

AWS OpsWorks Stacks menyimpan log internalnya di `/var/log/aws/opsworks` folder instance. Informasi ini biasanya tidak terlalu membantu untuk tujuan pemecahan masalah. Namun, log ini berguna untuk dukungan AWS OpsWorks Stacks, dan Anda mungkin diminta untuk menyediakannya jika Anda mengalami masalah dengan layanan. Log Linux juga terkadang dapat memberikan data pemecahan masalah yang berguna.

Instans Windows

Log Agen

Pada instance Windows, OpsWorks log disimpan di `ProgramData` jalur seperti berikut ini. Nomor tersebut termasuk stempel waktu.

```
C:\ProgramData\OpsWorksAgent\var\logs\number
```

> **Note**
>
> Secara default, `ProgramData` adalah folder tersembunyi. Untuk menampilkannya, navigasikan ke Opsi Folder. Di bawah View, pilih opsi untuk menampilkan file tersembunyi.

Contoh berikut menunjukkan log agen pada 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
```

Log Koki

Pada instance Windows, log Chef disimpan di `ProgramData` jalur seperti berikut ini. Nomor tersebut termasuk stempel waktu.

```
C:\ProgramData\OpsWorksAgent\var\commands\number
```

> **Note**
>
> Direktori ini hanya berisi output dari Chef run pertama (OpsWorks dimiliki).

Contoh berikut menunjukkan log Chef yang OpsWorks dimiliki pada 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
```

Log Chef Pengguna

Log untuk menjalankan Chef Anda dapat ditemukan dalam file bernama `logfile.txt` dalam folder yang dinamai sesuai perintah Chef bernomor, seperti pada diagram berikut.

C: /koki ---------- `runs` - `command-12345` `attribs.json` `client.rb` `logfile.txt`

Menafsirkan Log Chef

Awal log sebagian besar berisi logging Chef internal.

```
# 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
...
```

Bagian dari file ini sangat berguna bagi para ahli Chef. Perhatikan bahwa daftar run hanya mencakup dua resep, meskipun sebagian besar perintah melibatkan lebih banyak lagi. Kedua resep ini menangani tugas memuat dan mengeksekusi semua resep bawaan dan khusus lainnya.

Bagian yang paling menarik dari file biasanya di bagian akhir. Jika run berakhir dengan sukses, Anda akan melihat sesuatu seperti berikut:

```
...
[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**
>
> Anda dapat menggunakan agen CLI untuk menampilkan ekor log selama atau setelah dijalankan. Untuk informasi selengkapnya, lihat Menampilkan Log Chef.

Jika resep gagal, Anda harus mencari output tingkat Kesalahan, yang akan berisi pengecualian diikuti oleh jejak tumpukan Chef, seperti berikut ini:

```
...
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
```

Akhir file adalah jejak tumpukan Chef. Anda juga harus memeriksa output tepat sebelum pengecualian, yang sering berisi kesalahan sistem seperti `package not available` yang juga dapat berguna dalam menentukan penyebab kegagalan. Dalam hal ini, daemon MySQL gagal memulai.

Kesalahan Log Chef Umum

Berikut ini adalah beberapa kesalahan log Chef yang umum, dan cara mengatasinya.

Log tidak dapat ditemukan

Pada awal menjalankan Chef, instance menerima URL Amazon S3 yang telah ditetapkan sebelumnya yang memungkinkan Anda melihat log di halaman web saat Chef run selesai. Karena URL ini kedaluwarsa setelah dua jam, tidak ada log yang diunggah ke situs Amazon S3 jika Chef menjalankan membutuhkan waktu lebih dari dua jam, bahkan jika tidak ada masalah yang terjadi selama Chef menjalankan. Perintah untuk membuat log berhasil, tetapi log hanya dapat dilihat pada instance, bukan pada URL yang telah ditentukan sebelumnya.

Log berakhir tiba-tiba

Jika log Chef berakhir tiba-tiba tanpa menunjukkan keberhasilan atau menampilkan informasi kesalahan, Anda mungkin mengalami status memori rendah yang mencegah Chef menyelesaikan log. Pilihan terbaik Anda adalah mencoba lagi dengan contoh yang lebih besar.

Buku masak atau resep yang hilang

Jika Chef run menemukan buku masak atau resep yang tidak ada dalam cache buku masak, Anda akan melihat sesuatu seperti berikut:

```
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?
```

Entri ini menunjukkan bahwa `mycookbook` buku masak tidak ada dalam cache buku masak. Dengan Chef 11.4, Anda juga dapat mengalami kesalahan ini jika Anda tidak mendeklarasikan dependensi dengan benar. `metadata.rb`

AWS OpsWorks Stacks menjalankan resep dari cache buku masak instance. Ini mengunduh buku masak dari repositori Anda ke cache ini saat instance dimulai. Namun, AWS OpsWorks Stacks tidak secara otomatis memperbarui cache pada instance online jika Anda kemudian memodifikasi buku masak di repositori Anda. Jika Anda telah memodifikasi buku masak Anda atau menambahkan buku masak baru sejak memulai instance, ambil langkah-langkah berikut:

1. Pastikan bahwa Anda telah melakukan perubahan Anda ke repositori.
2. Jalankan perintah tumpukan Update Cookbooks untuk memperbarui cache buku masak dengan versi terbaru dari repositori.

Kegagalan perintah lokal

Jika `execute` sumber daya Chef gagal menjalankan perintah yang ditentukan, Anda akan melihat sesuatu seperti berikut:

```
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=/
```

Gulir ke atas di log dan Anda akan melihat perintah `stderr` dan `stdout` output, yang akan membantu Anda menentukan mengapa perintah gagal.

Kegagalan Package

Jika instalasi paket gagal, Anda akan melihat sesuatu seperti berikut:

```
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
```

Gulir ke atas di log dan Anda akan melihat output STDOUT dan STDERROR perintah, yang akan membantu Anda menentukan mengapa instalasi paket gagal.

Menggunakan AWS OpsWorks Stacks Agent CLI

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Note

Agen CLI hanya tersedia di instance Linux.

Pada setiap contoh online, AWS OpsWorks Stacks menginstal agen, yang berkomunikasi dengan layanan. Layanan AWS OpsWorks Stacks pada gilirannya mengirimkan perintah ke agen untuk melakukan tugas-tugas seperti memulai Chef berjalan pada instance ketika peristiwa siklus hidup terjadi. Pada instance Linux, agen mengekspos antarmuka baris perintah (CLI) yang sangat berguna untuk pemecahan masalah. Untuk menjalankan perintah CLI agen, gunakan SSH untuk terhubung ke sebuah instance. Anda kemudian dapat menjalankan perintah CLI agen untuk melakukan berbagai tugas, termasuk yang berikut ini:

- Jalankan resep.
- Tampilkan log Chef.
- Menampilkan konfigurasi tumpukan dan penerapan JSON.

Untuk informasi selengkapnya tentang cara mengatur koneksi SSH ke sebuah instance, lihatLogin dengan SSH. Anda juga harus memiliki izin SSH dan sudo untuk tumpukan.

Bagian ini menjelaskan cara menggunakan agen CLI untuk pemecahan masalah. Untuk informasi selengkapnya dan referensi perintah lengkap, lihatAWS OpsWorks Agen Tumpukan CLI.

Topik

- Mengeksekusi Resep
- Menampilkan Log Chef
- Menampilkan Konfigurasi Stack dan Deployment JSON

Mengeksekusi Resep

`run_command`Perintah agen CLI mengarahkan agen untuk menjalankan kembali perintah yang dilakukan sebelumnya. Perintah yang paling berguna untuk pemecahan masalah—`setup`,, `configuredeploy`, dan `undeploy` —masing-masing sesuai dengan peristiwa siklus hidup. Mereka mengarahkan agen untuk memulai lari Chef untuk mengeksekusi resep terkait.

> **Note**
>
> `run_command`Perintah ini terbatas untuk mengeksekusi kelompok resep yang dikaitkan dengan perintah tertentu, biasanya resep yang terkait dengan peristiwa siklus hidup. Anda tidak dapat menggunakannya untuk mengeksekusi resep tertentu. Untuk menjalankan satu atau beberapa resep yang ditentukan, gunakan perintah tumpukan Execute Recipes atau CLI yang setara atau tindakan API (`create-deployment`dan `CreateDeployment`).

`run_command`Perintah ini cukup berguna untuk men-debug resep khusus, terutama resep yang ditetapkan ke peristiwa siklus hidup Pengaturan dan Konfigurasi, yang tidak dapat Anda picu langsung dari konsol. Dengan menggunakan`run_command`, Anda dapat menjalankan resep acara tertentu sesering yang Anda butuhkan tanpa harus memulai atau menghentikan instance.

> **Note**
>
> AWS OpsWorks Stacks menjalankan resep dari cache buku masak instance, bukan repositori buku masak. AWS OpsWorks Tumpukan mengunduh buku masak ke cache ini saat instance dimulai, tetapi tidak secara otomatis memperbarui cache pada instance online jika Anda

kemudian memodifikasi buku masak Anda. Jika Anda telah memodifikasi buku masak Anda sejak memulai instance, pastikan untuk menjalankan perintah tumpukan perintah Update Cookbooks stack untuk memperbarui cache buku masak dengan versi terbaru dari repositori.

Agen hanya menyimpan perintah terbaru. Anda dapat mencantumkannya dengan menjalankan `list_commands`, yang mengembalikan daftar perintah yang di-cache dan waktu dilakukannya.

```
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
```

Untuk menjalankan kembali perintah terbaru, jalankan ini:

```
sudo opsworks-agent-cli run_command
```

Untuk menjalankan kembali instance terbaru dari perintah tertentu, jalankan ini:

```
sudo opsworks-agent-cli run_command command
```

Misalnya, untuk menjalankan kembali resep Setup, Anda dapat menjalankan perintah berikut:

```
sudo opsworks-agent-cli run_command setup
```

Setiap perintah memiliki konfigurasi tumpukan terkait dan penerapan JSON yang mewakili status tumpukan dan penerapan pada saat perintah dijalankan. Karena data itu dapat berubah dari satu perintah ke perintah berikutnya, instance perintah yang lebih lama mungkin menggunakan data yang agak berbeda dari yang terbaru. Untuk menjalankan kembali instance tertentu dari sebuah perintah, salin waktu dari `list_commands` output dan jalankan yang berikut:

```
sudo opsworks-agent-cli run_command time
```

Contoh sebelumnya semua menjalankan kembali perintah menggunakan JSON default, yang merupakan JSON diinstal untuk perintah itu. Anda dapat menjalankan kembali perintah terhadap file JSON arbitrer sebagai berikut:

```
sudo opsworks-agent-cli run_command -f /path/to/valid/json.file
```

Menampilkan Log Chef

`show_log`Perintah agen CLI menampilkan log tertentu. Setelah perintah selesai, Anda akan melihat akhir file. Oleh karena itu `show_log` perintah menyediakan cara yang mudah untuk mengekor log, yang biasanya di mana Anda menemukan informasi kesalahan. Anda dapat menggulir ke atas untuk melihat bagian log sebelumnya.

Untuk menampilkan log perintah saat ini, jalankan ini:

```
sudo opsworks-agent-cli show_log
```

Anda juga dapat menampilkan log untuk perintah tertentu, tetapi perlu diketahui bahwa agen cache log hanya untuk tiga puluh perintah terakhir. Anda dapat membuat daftar perintah instance dengan menjalankan `list_commands`, yang mengembalikan daftar perintah yang di-cache dan waktu dilakukannya. Sebagai contoh, lihat Mengeksekusi Resep.

Untuk menampilkan log untuk eksekusi terbaru dari perintah tertentu, jalankan yang berikut ini:

```
sudo opsworks-agent-cli show_log command
```

Parameter perintah dapat diatur `setup`,`configure`,`deploy`,`undeploy`,`start`,`stop`, atau`restart`. Sebagian besar perintah ini sesuai dengan peristiwa siklus hidup dan mengarahkan agen untuk menjalankan resep terkait.

Untuk menampilkan log untuk eksekusi perintah tertentu, salin tanggal dari `list_commands` output dan jalankan:

```
sudo opsworks-agent-cli show_log date
```

Jika perintah masih dijalankan, `show_log` menampilkan status log saat ini.

> **Note**
>
> Salah satu cara yang digunakan `show_log` untuk memecahkan masalah kesalahan dan out-of-memory masalah adalah dengan mengekor log selama eksekusi, sebagai berikut:

1. Gunakan `run_command` untuk memicu peristiwa siklus hidup yang sesuai. Untuk informasi selengkapnya, lihat Mengeksekusi Resep.
2. Berulang kali berlari `show_log` untuk melihat ekor log saat sedang ditulis.

Jika Chef kehabisan memori atau keluar secara tak terduga, log akan berakhir tiba-tiba. Jika resep gagal, log akan berakhir dengan pengecualian dan jejak tumpukan.

Menampilkan Konfigurasi Stack dan Deployment JSON

Sebagian besar data yang digunakan oleh resep berasal dari konfigurasi tumpukan dan penerapan JSON, yang mendefinisikan satu set atribut Chef yang memberikan deskripsi rinci tentang konfigurasi tumpukan, penerapan apa pun, dan atribut kustom opsional yang dapat ditambahkan pengguna. Untuk setiap perintah, AWS OpsWorks Stacks menginstal JSON yang mewakili status tumpukan dan penerapan pada saat perintah. Untuk informasi selengkapnya, lihat Konfigurasi Stack dan Atribut Deployment.

Jika resep kustom Anda memperoleh data dari konfigurasi tumpukan dan penerapan JSON, Anda dapat memverifikasi data dengan memeriksa JSON. Cara termudah untuk menampilkan konfigurasi tumpukan dan penyebaran JSON adalah dengan menjalankan `get_json`perintah agen CLI, yang menampilkan versi diformat dari objek JSON. Berikut ini menunjukkan beberapa baris pertama dari beberapa output tipikal:

```
{
  "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"
...
```

Anda dapat menampilkan konfigurasi tumpukan terbaru dan penyebaran JSON sebagai berikut:

```
sudo opsworks-agent-cli get_json
```

Anda dapat menampilkan konfigurasi tumpukan terbaru dan penyebaran JSON untuk perintah tertentu dengan menjalankan yang berikut:

```
sudo opsworks-agent-cli get_json command
```

Parameter perintah dapat diatur `setup`,`configure`,`deploy`,`undeploy`,`start`,`stop`, atau`restart`. Sebagian besar perintah ini sesuai dengan peristiwa siklus hidup dan mengarahkan agen untuk menjalankan resep terkait.

Anda dapat menampilkan konfigurasi tumpukan dan penerapan JSON untuk eksekusi perintah tertentu dengan menentukan tanggal perintah seperti ini:

```
sudo opsworks-agent-cli get_json date
```

Cara paling sederhana untuk menggunakan perintah ini adalah sebagai berikut:

1. Jalankan`list_commands`, yang mengembalikan daftar perintah yang telah dijalankan pada instance, dan tanggal setiap perintah dijalankan.
2. Salin tanggal untuk perintah yang sesuai dan gunakan sebagai `get_json` *`date`* argumen.

Masalah Debugging dan Pemecahan Masalah Umum

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini menjelaskan beberapa masalah debugging dan pemecahan masalah yang umum ditemui serta solusinya.

Topik

- Pemecahan Masalah Tumpukan AWS OpsWorks
- Pemecahan Masalah Pendaftaran Instans

Pemecahan Masalah Tumpukan AWS OpsWorks

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Bagian ini berisi beberapa masalah AWS OpsWorks Stacks yang umum ditemui dan solusinya.

Topik

- Tidak Dapat Mengelola Instance
- Setelah Chef Run, Instance Tidak Akan Boot
- Contoh Lapisan Semua Gagal Pemeriksaan Kesehatan Elastic Load Balancing
- Tidak Dapat Berkomunikasi dengan Load Balancer Elastic Load Balancing
- Instance Lokal yang Diimpor Gagal Menyelesaikan Pengaturan Volume Setelah Mulai Ulang
- Volume EBS Tidak Terpasang Kembali Setelah Reboot
- Tidak Dapat Menghapus AWS OpsWorks Grup Keamanan Stacks
- Tidak Sengaja Menghapus Grup Keamanan AWS OpsWorks Stacks
- Chef Log Berakhir Tiba-tiba

- Cookbook Tidak Diperbarui
- Instance Terjebak di Status Booting
- Instans Secara Tak Terduga Mulai Ulang
- opsworks-agentProses Berjalan pada Instance
- Perintah execute_recipes yang tidak terduga

Tidak Dapat Mengelola Instance

Masalah: Anda tidak lagi dapat mengelola instance yang telah dikelola di masa lalu. Dalam beberapa kasus, log dapat menunjukkan kesalahan yang mirip dengan yang berikut ini.

```
Aws::CharlieInstanceService::Errors::UnrecognizedClientException - The security token
 included in the request is invalid.
```

Penyebab: Ini dapat terjadi jika sumber daya AWS OpsWorks di luar tempat instance bergantung telah diedit atau dihapus. Berikut ini adalah contoh perubahan sumber daya yang dapat memutus komunikasi dengan sebuah instance.

- Pengguna IAM atau peran yang terkait dengan instance telah dihapus secara tidak sengaja, di luar AWS OpsWorks Stacks. Hal ini menyebabkan kegagalan komunikasi antara AWS OpsWorks agen yang diinstal pada instance, dan layanan AWS OpsWorks Stacks. Pengguna yang terkait dengan instance diperlukan sepanjang umur instance.
- Mengedit konfigurasi volume atau penyimpanan saat instance offline dapat membuat instance tidak dapat dikelola.
- Menambahkan EC2 instance ke ELB secara manual. AWS OpsWorks mengonfigurasi ulang penyeimbang beban Elastic Load Balancing yang ditetapkan setiap kali instance masuk atau keluar dari status online. AWS OpsWorks hanya menganggap contoh yang diketahuinya akan menjadi anggota yang valid; contoh yang ditambahkan di luar AWS OpsWorks, atau oleh beberapa proses lain, dihapus. Setiap contoh lainnya dihapus.

Solusi: Jangan hapus pengguna IAM atau peran yang menjadi sandaran instans Anda. Jika memungkinkan, edit konfigurasi volume atau penyimpanan hanya saat instance dependen sedang berjalan. Gunakan AWS OpsWorks untuk mengelola load balancer atau keanggotaan EIP instance. AWS OpsWorks Saat Anda mendaftarkan instance, untuk membantu mencegah masalah dalam mengelola instance terdaftar jika pengguna terhapus secara tidak sengaja, tambahkan `--use-`

`instance-profile` parameter ke `register` perintah Anda untuk menggunakan profil instans bawaan instans sebagai gantinya.

Setelah Chef Run, Instance Tidak Akan Boot

Masalah: Pada Chef 11.10 atau tumpukan yang lebih lama yang dikonfigurasi untuk menggunakan buku masak khusus, setelah Chef menjalankan yang menggunakan buku masak komunitas, instance tidak akan boot. Pesan log dapat menyatakan bahwa resep gagal dikompilasi ("Kesalahan Kompilasi Resep"), atau tidak dapat dimuat karena tidak dapat menemukan ketergantungan.

Penyebab: Penyebab yang paling mungkin adalah buku masak kustom atau komunitas tidak mendukung versi Chef yang digunakan tumpukan Anda. Beberapa buku masak komunitas populer, seperti `apt` dan`build-essential`, telah mengetahui masalah kompatibilitas dengan Chef 11.10.

Solusi: Pada AWS OpsWorks tumpukan Tumpukan yang mengaktifkan pengaturan Gunakan buku masak Chef kustom, buku masak khusus atau komunitas harus selalu mendukung versi Chef yang digunakan tumpukan Anda. Sematkan buku masak komunitas ke versi (yaitu, atur nomor versi buku masak ke versi tertentu) yang kompatibel dengan versi Chef yang dikonfigurasi dalam pengaturan tumpukan Anda. Untuk menemukan versi buku masak komunitas yang didukung, lihat changelog untuk buku masak yang gagal dikompilasi, dan gunakan hanya versi terbaru dari buku masak yang akan didukung oleh tumpukan Anda. Untuk menyematkan versi buku masak, tentukan nomor versi yang tepat di Berksfile repositori buku masak kustom Anda. Misalnya, `cookbook 'build-essential', '= 3.2.0'`.

Contoh Lapisan Semua Gagal Pemeriksaan Kesehatan Elastic Load Balancing

Masalah: Anda melampirkan penyeimbang beban Elastic Load Balancing ke lapisan server aplikasi, tetapi semua instance gagal dalam pemeriksaan kesehatan.

Penyebab: Saat membuat penyeimbang beban Elastic Load Balancing, Anda harus menentukan jalur ping yang dipanggil penyeimbang beban untuk menentukan apakah instans tersebut sehat. Pastikan untuk menentukan jalur ping yang sesuai untuk aplikasi Anda; nilai default adalah /index.html. Jika aplikasi Anda tidak termasuk`index.html`, Anda harus menentukan jalur yang sesuai atau pemeriksaan kesehatan akan gagal. Misalnya, PHPApp aplikasi Simple yang digunakan Memulai dengan Chef 11 Linux Stacks tidak menggunakan index.html; jalur ping yang sesuai untuk server tersebut adalah /.

Solusi: Edit jalur ping penyeimbang beban. Untuk informasi selengkapnya, lihat Elastic Load Balancing

Tidak Dapat Berkomunikasi dengan Load Balancer Elastic Load Balancing

Masalah: Anda membuat penyeimbang beban Elastic Load Balancing dan melampirkannya ke lapisan server aplikasi, tetapi ketika Anda mengklik nama DNS atau alamat IP penyeimbang beban untuk menjalankan aplikasi, Anda mendapatkan kesalahan berikut: “Server jarak jauh tidak merespons”.

Penyebab: Jika tumpukan Anda berjalan di VPC default, saat Anda membuat penyeimbang beban Elastic Load Balancing di wilayah tersebut, Anda harus menentukan grup keamanan. Grup keamanan harus memiliki aturan masuk yang memungkinkan lalu lintas masuk dari alamat IP Anda. Jika Anda menentukan grup keamanan VPC default, aturan masuk default tidak menerima lalu lintas masuk apa pun.

Solusi: Edit aturan masuk grup keamanan untuk menerima lalu lintas masuk dari alamat IP yang sesuai.

1. Klik Grup Keamanan di panel navigasi EC2 konsol Amazon.
2. Pilih grup keamanan load balancer.
3. Klik Edit pada tab Inbound.
4. Tambahkan aturan masuk dengan Sumber disetel ke CIDR yang sesuai.

   Misalnya, menentukan Anywhere menetapkan CIDR ke 0.0.0.0/0, yang mengarahkan penyeimbang beban untuk menerima lalu lintas masuk dari alamat IP apa pun.

Instance Lokal yang Diimpor Gagal Menyelesaikan Pengaturan Volume Setelah Mulai Ulang

Masalah: Anda memulai ulang EC2 instance yang telah diimpor ke AWS OpsWorks Stacks, dan tampilan konsol AWS OpsWorks Stacks gagal sebagai status instans. Ini dapat terjadi pada contoh Chef 11 atau Chef 12.

Penyebab:AWS OpsWorks Tumpukan mungkin tidak dapat melampirkan volume ke instance Anda selama proses penyiapan. Salah satu kemungkinan penyebabnya adalah AWS OpsWorks Stacks menimpa konfigurasi volume Anda pada instance Anda saat Anda menjalankan perintah. `setup`

Solusi: Buka halaman Detail untuk instance, dan periksa konfigurasi volume Anda di area Volume. Perhatikan bahwa Anda dapat mengubah konfigurasi volume hanya jika instans Anda berada dalam status berhenti. Pastikan bahwa setiap volume memiliki titik pemasangan dan nama yang ditentukan. Konfirmasikan bahwa Anda memberikan titik pemasangan yang benar dalam konfigurasi di AWS OpsWorks Stacks sebelum memulai ulang instance.

Volume EBS Tidak Terpasang Kembali Setelah Reboot

Masalah: Anda menggunakan EC2 konsol Amazon untuk melampirkan volume Amazon EBS ke instans tetapi ketika Anda me-reboot instance, volume tidak lagi terpasang.

Penyebab: AWS OpsWorks Tumpukan hanya dapat memasang kembali volume Amazon EBS yang disadarinya, yang terbatas pada hal-hal berikut:

- Volume yang dibuat oleh AWS OpsWorks Stacks.
- Volume dari akun Anda yang telah Anda daftarkan secara eksplisit dengan tumpukan menggunakan halaman Resources.

Solusi: Kelola volume Amazon EBS Anda hanya dengan menggunakan konsol AWS OpsWorks Stacks, API, atau CLI. Jika Anda ingin menggunakan salah satu volume Amazon EBS akun Anda dengan tumpukan, gunakan halaman Sumber Daya tumpukan untuk mendaftarkan volume dan melampirkannya ke sebuah instance. Untuk informasi selengkapnya, lihat Manajemen Sumber Daya.

Tidak Dapat Menghapus AWS OpsWorks Grup Keamanan Stacks

Masalah: Setelah Anda menghapus tumpukan, ada sejumlah grup keamanan AWS OpsWorks Stacks yang tertinggal yang tidak dapat dihapus.

Penyebab: Grup keamanan harus dihapus dalam urutan tertentu.

Solusi: Pertama, pastikan tidak ada instance yang menggunakan grup keamanan. Kemudian, hapus salah satu grup keamanan berikut, jika ada, dengan urutan sebagai berikut:

1. AWS- OpsWorks -Server Kosong
2. AWS- OpsWorks -Monitoring-Master-Server
3. AWS- OpsWorks -DB-Master-Server
4. AWS- OpsWorks -Memcached-Server
5. AWS- OpsWorks -Server Khusus
6. AWS- OpsWorks -NodeJS-App-Server
7. AWS- OpsWorks -PHP-APP-Server
8. AWS- OpsWorks -Rails-App-Server
9. AWS- OpsWorks -Server Web
10. AWS- OpsWorks -Server Default

11AWS- OpsWorks -LB-Server

Tidak Sengaja Menghapus Grup Keamanan AWS OpsWorks Stacks

Masalah: Anda menghapus salah satu grup keamanan AWS OpsWorks Stacks dan perlu membuatnya kembali.

Penyebab: Grup keamanan ini biasanya dihapus secara tidak sengaja.

Solusi: Grup yang dibuat ulang harus duplikat persis dari aslinya, termasuk kapitalisasi yang sama untuk nama grup. Alih-alih membuat ulang grup secara manual, pendekatan yang lebih disukai adalah meminta AWS OpsWorks Stacks melakukan tugas untuk Anda. Cukup buat tumpukan baru di wilayah AWS yang sama—dan VPC, jika AWS OpsWorks ada—dan Stacks akan secara otomatis membuat ulang semua grup keamanan bawaan, termasuk yang Anda hapus. Anda kemudian dapat menghapus tumpukan jika Anda tidak memiliki penggunaan lebih lanjut untuk itu; grup keamanan akan tetap ada.

Chef Log Berakhir Tiba-tiba

Masalah: Log Chef berakhir secara tiba-tiba; akhir log tidak menunjukkan keberhasilan menjalankan atau menampilkan pengecualian dan jejak tumpukan.

Penyebab: Perilaku ini biasanya disebabkan oleh memori yang tidak memadai.

Solusi: Buat instance yang lebih besar dan gunakan `run_command` perintah CLI agen untuk menjalankan resep lagi. Untuk informasi selengkapnya, lihat Mengeksekusi Resep.

Cookbook Tidak Diperbarui

Masalah: Anda memperbarui buku masak Anda tetapi instance tumpukan masih menjalankan resep lama.

Penyebab: AWS OpsWorks Tumpukan menyimpan buku masak di setiap instance, dan menjalankan resep dari cache, bukan repositori. Saat Anda memulai instance baru, AWS OpsWorks Stacks mengunduh buku masak Anda dari repositori ke cache instance. Namun, jika Anda kemudian memodifikasi buku masak khusus Anda, AWS OpsWorks Stacks tidak secara otomatis memperbarui cache instans online.

Solusi: Jalankan perintah tumpukan Update Cookbooks untuk secara eksplisit mengarahkan AWS OpsWorks Stacks untuk memperbarui cache buku masak instans online Anda.

Instance Terjebak di Status Booting

Masalah: Ketika Anda me-restart sebuah instance, atau auto healing restart secara otomatis, operasi startup berhenti pada `booting` status.

Penyebab: Salah satu kemungkinan penyebab masalah ini adalah konfigurasi VPC, termasuk VPC default. Instans harus selalu dapat berkomunikasi dengan layanan AWS OpsWorks Stacks, Amazon S3, dan paket, buku masak, dan repositori aplikasi. Jika, misalnya, Anda menghapus gateway default dari VPC default, instance kehilangan koneksinya ke layanan Stacks. AWS OpsWorks Karena AWS OpsWorks Stacks tidak dapat lagi berkomunikasi dengan agen instance, ia memperlakukan instance sebagai gagal dan menyembuhkannya secara otomatis. Namun, tanpa koneksi, AWS OpsWorks Stacks tidak dapat menginstal agen instance pada instance yang disembuhkan. Tanpa agen, AWS OpsWorks Stacks tidak dapat menjalankan resep Pengaturan pada instance, sehingga operasi startup tidak dapat berkembang melampaui status “booting”.

Solusi: Ubah konfigurasi VPC Anda sehingga instance memiliki konektivitas yang diperlukan.

Instans Secara Tak Terduga Mulai Ulang

Masalah: Instance yang berhenti tiba-tiba dimulai ulang.

Penyebab 1: Jika Anda telah mengaktifkan penyembuhan otomatis untuk instans Anda, AWS OpsWorks Stacks secara berkala melakukan pemeriksaan kesehatan pada EC2 instans Amazon terkait, dan memulai ulang yang tidak sehat. Jika Anda menghentikan atau menghentikan instance yang AWS OpsWorks dikelola Stacks menggunakan EC2 konsol Amazon, API, atau CLI, Stacks tidak akan diberi tahu. AWS OpsWorks Sebaliknya, ia akan menganggap instance yang dihentikan sebagai tidak sehat dan secara otomatis memulai ulang.

Solusi: Kelola instans Anda hanya dengan menggunakan konsol AWS OpsWorks Stacks, API, atau CLI. Jika Anda menggunakan AWS OpsWorks Stacks untuk menghentikan atau menghapus instance, itu tidak akan dimulai ulang. Untuk informasi selengkapnya, silakan lihat Memulai, Menghentikan, dan Memulai Ulang Instans 24/7 Secara Manual dan Menghapus Instans AWS OpsWorks Stacks.

Penyebab 2: Contoh dapat gagal karena berbagai alasan. Jika Anda mengaktifkan penyembuhan otomatis, AWS OpsWorks Stacks secara otomatis memulai ulang instance yang gagal.

Solusi: Ini adalah operasi normal; tidak perlu melakukan apa pun kecuali Anda tidak ingin AWS OpsWorks Stacks memulai ulang instance yang gagal. Dalam hal ini, Anda harus menonaktifkan penyembuhan otomatis.

`opsworks-agent` Proses Berjalan pada Instance

Masalah: Beberapa `opsworks-agent` proses berjalan pada instance Anda. Sebagai contoh:

```
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
```

Penyebab: Ini adalah proses sah yang diperlukan untuk operasi normal agen. Mereka melakukan tugas-tugas seperti menangani penerapan dan mengirim pesan keep-alive kembali ke layanan.

Solusi: Ini adalah perilaku normal. Jangan hentikan proses ini; melakukannya akan membahayakan operasi agen.

Perintah execute_recipes yang tidak terduga

Masalah: Bagian Log pada halaman detail instance menyertakan `execute_recipes` perintah yang tidak terduga. `execute_recipes`Perintah tak terduga juga dapat muncul di halaman Stack dan Deployment.

Penyebab: Masalah ini sering disebabkan oleh perubahan izin. Saat Anda mengubah izin SSH atau sudo pengguna atau grup, AWS OpsWorks Stacks berjalan `execute_recipes` untuk memperbarui instance tumpukan dan juga memicu peristiwa Konfigurasi. Sumber `execute_recipes` perintah lainnya adalah AWS OpsWorks Stacks memperbarui agen instance.

Solusi: Ini adalah operasi normal; tidak perlu melakukan apa pun.

Untuk melihat tindakan apa yang dilakukan `execute_recipes` perintah, buka halaman Deployments dan klik stempel waktu perintah. Ini membuka halaman detail perintah, yang mencantumkan resep utama yang dijalankan. Misalnya, halaman detail berikut adalah untuk `execute_recipes` perintah yang dijalankan `ssh_users` untuk memperbarui izin SSH.



Untuk melihat semua detail, klik tampilkan di kolom Log perintah untuk menampilkan log Chef terkait. Cari log untuk**`Run List`**. AWS OpsWorks Resep pemeliharaan tumpukan akan berada di bawah`OpsWorks Custom Run List`. Misalnya, berikut ini adalah daftar run untuk `execute_recipes` perintah yang ditunjukkan pada tangkapan layar sebelumnya, dan menunjukkan setiap resep yang terkait dengan perintah.

```
[2014-02-21T17:16:30+00:00] INFO: OpsWorks Custom Run List:
 ["opsworks_stack_state_sync",
  "ssh_users", "test_suite", "opsworks_cleanup"]
```

Pemecahan Masalah Pendaftaran Instans

Bagian ini berisi beberapa masalah pendaftaran instance yang umum ditemui dan solusinya.

> **Note**
>
> Jika Anda mengalami masalah pendaftaran, jalankan `register` dengan `--debug` argumen, yang memberikan informasi debugging tambahan.

Topik

- EC2Pengguna Tidak Berwenang untuk Melakukan:...
- Kredensi Harus Dicakup ke Wilayah yang Valid

EC2Pengguna Tidak Berwenang untuk Melakukan:...

Masalah: Sebuah `register` perintah mengembalikan sesuatu seperti berikut:

```
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
```

Penyebab: `register` Perintah berjalan dengan kredensil yang tidak memberikan izin yang diperlukan. Kebijakan pengguna harus mengizinkan `iam:CreateGroup` tindakan, antara lain.

Solusi Menyediakan `register` kredensil pengguna IAM yang memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat Menginstal dan Mengkonfigurasi AWS CLI.

Kredensi Harus Dicakup ke Wilayah yang Valid

Masalah: Sebuah `register` perintah mengembalikan yang berikut:

```
A client error (InvalidSignatureException) occurred when calling the
DescribeStacks operation: Credential should be scoped to a valid region, not 'cn-
north-1'.
```

Penyebab: Wilayah perintah harus merupakan wilayah AWS OpsWorks Stacks yang valid. Untuk daftar wilayah yang didukung, lihatSupport Wilayah. Kesalahan ini biasanya terjadi karena salah satu alasan berikut:

- Tumpukan berada di wilayah yang berbeda, dan Anda menetapkan wilayah tumpukan ke `--region` argumen perintah.

  Anda tidak perlu menentukan wilayah tumpukan; AWS OpsWorks Tumpukan secara otomatis menentukannya dari ID tumpukan.
- Anda menghilangkan `--region` argumen, yang secara implisit menentukan wilayah default, tetapi wilayah default Anda tidak didukung oleh Stacks. AWS OpsWorks

Solusi: Setel secara eksplisit `--region` ke wilayah AWS OpsWorks Stacks yang didukung, atau edit AWS CLI `config` file Anda untuk mengubah wilayah default ke wilayah Stacks yang didukung AWS OpsWorks . Untuk informasi selengkapnya, lihat Mengonfigurasi Antarmuka Baris Perintah AWS.

AWS OpsWorks Agen Tumpukan CLI

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

> **Note**
>
> Fitur ini hanya tersedia di instance Linux.

Agen yang diinstal AWS OpsWorks Stacks pada setiap instance mengekspos antarmuka baris perintah (CLI). Jika Anda menggunakan SSH untuk masuk ke instance, Anda dapat menggunakan CLI sebagai berikut:

- Akses file log untuk Chef berjalan.
- Akses perintah AWS OpsWorks Stacks.
- Jalankan resep Chef secara manual.
- Lihat laporan contoh.
- Lihat laporan agen.
- Lihat satu set konfigurasi tumpukan dan atribut penerapan terbatas.

> **Important**
>
> Anda dapat menjalankan perintah CLI agen hanya sebagai root atau dengan menggunakan. `sudo`

Sintaks perintah dasar adalah:

```
sudo opsworks-agent-cli [--help] [command [activity] [date]]
```

Keempat argumen tersebut adalah sebagai berikut:

membantu

(Opsional) Menampilkan sinopsis singkat dari perintah yang tersedia saat digunakan dengan sendirinya. Saat digunakan dengan perintah, `help` menampilkan deskripsi perintah.

perintah

(Opsional) Perintah agen CLI, yang harus diatur ke salah satu dari berikut ini:

- agent_report
- get_json
- instance_report
- list_commands
- run_command
- show_log
- stack_state

aktivitas

(Opsional) Digunakan sebagai argumen dengan beberapa perintah untuk menentukan aktivitas AWS OpsWorks Stacks tertentu:`setup`,,`configure`,`deploy`,`undeploy`, `startstop`, ataurestart.

tanggal

(Opsional) Digunakan sebagai argumen dengan beberapa perintah untuk menentukan eksekusi perintah AWS OpsWorks Stacks tertentu. Tentukan eksekusi perintah dengan mengatur tanggal ke stempel waktu bahwa perintah dijalankan dalam *`yyyy-mm-ddThh:mm:ss`* format, termasuk tanda kutip tunggal. Misalnya, untuk 10:31:55 pada hari Selasa 5 Februari 2013, gunakan:. `'2013-02-05T10:31:55'` Untuk menentukan kapan perintah AWS OpsWorks Stacks tertentu dijalankan, jalankan list_commands.

> **Note**
>
> Jika agen telah mengeksekusi aktivitas AWS OpsWorks Stacks yang sama beberapa kali, Anda dapat memilih eksekusi tertentu dengan menentukan aktivitas dan waktu pelaksanaannya. Jika Anda menentukan aktivitas dan menghilangkan waktu, perintah CLI

agen akan bertindak pada eksekusi terbaru aktivitas tersebut. Jika Anda menghilangkan kedua argumen, perintah agen CLI bertindak pada aktivitas terbaru.

Bagian berikut menjelaskan perintah dan argumen yang terkait. Untuk singkatnya, bagian sintaks menghilangkan `--help` opsi opsional, yang dapat digunakan dengan perintah apa pun.

Topik

- agent_report
- get_json
- instance_report
- list_commands
- run_command
- show_log
- stack_state

agent_report

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Mengembalikan laporan agen.

```
sudo opsworks-agent-cli agent_report
```

Contoh keluaran berikut adalah dari instance yang baru-baru ini menjalankan aktivitas konfigurasi.

```
$ 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**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Mengembalikan informasi tentang Chef menjalankan sebagai objek JSON.

```
sudo opsworks-agent-cli get_json [activity] [date] [-i | --internal | --no-i | --no-internal]
```

Secara default, `get_json` menampilkan informasi yang disediakan pelanggan untuk menjalankan Chef terbaru. Gunakan opsi berikut untuk menentukan kumpulan informasi tertentu.

aktivitas

Menampilkan informasi untuk menjalankan Chef yang terkait dengan aktivitas tertentu terbaru. Untuk mendapatkan daftar kegiatan yang valid, jalankan list_commands.

tanggal

Menampilkan informasi untuk menjalankan Chef yang terkait dengan aktivitas yang dijalankan untuk stempel waktu yang ditentukan. Untuk mendapatkan daftar stempel waktu yang valid, jalankan. list_commands

-i, --internal

Menampilkan informasi yang digunakan AWS OpsWorks Stacks secara internal untuk menjalankan Chef.

--tidak-i, --tidak-internal

Secara eksplisit menampilkan informasi yang disediakan pelanggan untuk menjalankan Chef. Ini adalah default jika tidak ditentukan lain.

> **Note**
>
> Untuk instance Chef 12 Linux, menjalankan perintah ini akan mengembalikan informasi yang valid seperti konfigurasi stack instance dan atribut deployment. Namun, untuk mendapatkan informasi yang lebih lengkap, rujuk kantong data Chef yang dibuat AWS OpsWorks Stacks pada instance. Untuk informasi selengkapnya, lihat AWS OpsWorks Referensi Tas Data Stacks.

Contoh keluaran berikut menunjukkan informasi yang disediakan pelanggan untuk Chef terbaru yang dijalankan untuk aktivitas konfigurasi terbaru.

```
$ sudo opsworks-agent-cli get_json configure

{
  "run_list": [
    "recipe[opsworks_cookbook_demo::configure]"
  ]
}
```

Contoh keluaran berikut menunjukkan informasi yang digunakan AWS OpsWorks Stacks secara internal untuk menjalankan Chef yang dijalankan untuk stempel waktu yang ditentukan.

```
$ 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**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Mengembalikan laporan contoh diperpanjang.

```
sudo opsworks-agent-cli instance_report
```

Contoh output berikut adalah dari sebuah 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**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika

Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Daftar waktu untuk setiap aktivitas yang telah dijalankan pada instance ini. Anda dapat menggunakan waktu-waktu ini untuk perintah agen-CLI lainnya untuk menentukan eksekusi tertentu.

```
sudo opsworks-agent-cli list_commands  [activity] [date]
```

Contoh keluaran berikut adalah dari instance yang telah menjalankan konfigurasi, penyiapan, dan memperbarui aktivitas buku masak kustom.

```
$ 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

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Menjalankan perintah AWS OpsWorks Stacks, yang merupakan file JSON yang berisi run-list Chef yang berisi informasi yang diperlukan untuk menjalankan aktivitas AWS OpsWorks Stacks (setup, configure, deploy, dan sebagainya). `run_command`Perintah menghasilkan entri log yang dapat Anda lihat dengan menjalankan show_log . Opsi ini dimaksudkan hanya untuk tujuan pengembangan, sehingga AWS OpsWorks Stacks tidak melacak perubahan.

```
sudo opsworks-agent-cli run_command [activity] [date] [/path/to/valid/json.file]
```

Secara default, `run_command` jalankan perintah AWS OpsWorks Stacks terbaru. Gunakan opsi berikut untuk menentukan perintah tertentu.

aktivitas

Jalankan perintah AWS OpsWorks Stacks tertentu:`setup`,`configure`,`deploy`,`undeploy`, `startstop`, atau`restart`.

tanggal

Jalankan OpsWorks perintah AWS yang dijalankan pada stempel waktu yang ditentukan. Untuk mendapatkan daftar stempel waktu yang valid, jalankan. list_commands

berkas

Jalankan file JSON perintah yang ditentukan. Untuk mendapatkan path file perintah, jalankan get_json.

Contoh output berikut adalah dari sebuah instance dan menjalankan perintah 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**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Return adalah file log perintah.

```
sudo opsworks-agent-cli show_log [activity] [date]
```

Secara default, `show_log` buntut file log terbaru. Gunakan opsi berikut untuk menentukan perintah tertentu.

aktivitas

Menampilkan file log aktivitas yang ditentukan.

tanggal

Menampilkan file log untuk aktivitas yang dijalankan pada stempel waktu yang ditentukan. Untuk mendapatkan daftar stempel waktu yang valid, jalankan. `list_commands`

Contoh keluaran berikut menunjukkan log terbaru.

```
$ 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**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Menampilkan informasi yang digunakan AWS OpsWorks Stacks secara internal untuk menjalankan Chef terbaru.

```
opsworks-agent-cli stack_state
```

> **Note**
>
> Untuk instance Chef 12 Linux, menjalankan perintah ini akan mengembalikan informasi yang valid seperti konfigurasi stack instance dan atribut deployment. Namun, untuk mendapatkan informasi yang lebih lengkap, rujuk kantong data Chef yang dibuat AWS OpsWorks Stacks pada instance. Untuk informasi selengkapnya, lihat AWS OpsWorks Referensi Tas Data Stacks.

Contoh output berikut adalah dari sebuah 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 Referensi Tas Data Stacks

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

AWS OpsWorks Stacks memaparkan berbagai macam pengaturan ke resep sebagai konten kantong data Chef. Referensi ini mencantumkan konten tas data ini.

Data bag adalah konsep Chef. Data bag adalah variabel global yang disimpan sebagai data JSON pada sebuah instance; data JSON dapat diakses dari Chef. Misalnya, tas data dapat menyimpan variabel global seperti URL sumber aplikasi, nama host instance, dan pengidentifikasi VPC stack terkait. AWS OpsWorks Stacks menyimpan tas datanya pada setiap instance tumpukan. Pada instance Linux, AWS OpsWorks Stacks menyimpan kantong data di direktori. `/var/chef/runs/`*`run-ID`*`/data_bags` Pada instance Windows, ia menyimpan tas data di *`drive`*`:\chef\runs\`*`run-id`*`\data_bags` direktori. Dalam kedua kasus, *`run-ID`* adalah ID unik yang ditetapkan

AWS OpsWorks Stacks ke setiap Chef yang dijalankan pada sebuah instance. Direktori ini mencakup satu set tas data (subdirektori). Setiap tas data berisi nol atau lebih item tas data, yang merupakan file berformat JSON yang berisi kumpulan konten tas data.

> **Note**
>
> AWS OpsWorks Stacks tidak mendukung kantong data terenkripsi. Untuk menyimpan data sensitif dalam bentuk terenkripsi, seperti kata sandi atau sertifikat, sebaiknya simpan di bucket S3 pribadi. Anda kemudian dapat membuat resep khusus yang menggunakan Amazon SDK for Ruby untuk mengambil data. Sebagai contoh, lihat Menggunakan SDK untuk Ruby.

Isi tas data dapat mencakup salah satu dari yang berikut:

- Konten string yang mengikuti sintaks Ruby standar dan dapat menggunakan tanda kutip tunggal atau ganda, meskipun string yang berisi karakter khusus tertentu harus memiliki tanda kutip ganda. Untuk informasi lebih lanjut, kunjungi situs dokumentasi Ruby.
- Konten Boolean, yang merupakan salah satu `true` atau `false` (tidak ada tanda kutip).
- Konten angka, yang merupakan bilangan bulat atau angka desimal, seperti `4` atau `2.5` (tidak ada tanda kutip).
- Daftar konten, yang mengambil bentuk nilai dipisahkan koma tertutup dalam tanda kurung siku (tanpa tanda kutip), seperti `[ '80', '443' ]`
- Objek JSON, yang berisi konten tas data tambahan, seperti`"my-app": {"elastic_ip": null,...}`.

Resep koki dapat mengakses tas data, item tas data, dan konten tas data melalui pencarian Chef atau langsung. Berikut ini menjelaskan cara menggunakan kedua pendekatan akses (meskipun pencarian Chef lebih disukai).

Untuk mengakses tas data melalui pencarian Chef, gunakan metode pencarian, tentukan indeks pencarian yang diinginkan. AWS OpsWorks Stacks menyediakan indeks pencarian berikut:

- aws_opsworks_app, yang mewakili satu set aplikasi yang diterapkan untuk tumpukan.
- aws_opsworks_command, yang mewakili satu set perintah yang dijalankan pada tumpukan.
- aws_opsworks_ecs_cluster, yang mewakili satu set instance cluster Amazon Elastic Container Service (Amazon ECS) Container Service (Amazon ECS) untuk tumpukan.

- aws_opsworks_elastic_load_balancer, yang mewakili satu set penyeimbang beban Elastic Load Balancing untuk tumpukan.
- aws_opsworks_instance, yang mewakili satu set instance untuk tumpukan.
- aws_opsworks_layer, yang mewakili satu set lapisan untuk tumpukan.
- aws_opsworks_rds_db_instance, yang mewakili sekumpulan instance Amazon Relational Database Service (Amazon RDS) untuk tumpukan.
- aws_opsworks_stack, yang mewakili tumpukan.
- aws_opsworks_user, yang mewakili satu set pengguna untuk tumpukan.

Setelah Anda mengetahui nama indeks pencarian, Anda dapat mengakses konten tas data untuk indeks pencarian tersebut. Misalnya, kode resep berikut menggunakan indeks `aws_opsworks_app` pencarian untuk mendapatkan konten item tas data pertama (file JSON pertama) di tas `aws_opsworks_app` data (`aws_opsworks_app`direktori). Kode kemudian menulis dua pesan ke log Chef, satu dengan konten tas data shortname aplikasi (string dalam file JSON), dan satu lagi dengan konten tas data URL sumber aplikasi (string lain dalam file JSON):

```
app = search("aws_opsworks_app").first
Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}' **********")
```

Di mana `['shortname']` dan `['app_source']['url']` tentukan konten tas data berikut dalam file JSON yang sesuai:

```
{
  ...
  "shortname": "mylinuxdemoapp",
  ...
  "app_source": {
    ...
    "url": "https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-
nodejs.tar.gz",
  },
  ...
}
```

Untuk daftar konten tas data yang dapat Anda cari, lihat topik referensi di bagian ini.

Anda juga dapat mengulangi melalui satu set item tas data dalam tas data. Misalnya, kode resep berikut mirip dengan contoh sebelumnya; itu berulang melalui masing-masing item tas data di tas data ketika ada lebih dari satu item tas data:

```
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
```

Jika Anda tahu bahwa konten tas data tertentu ada, Anda dapat menemukan item tas data yang sesuai dengan sintaks berikut:

```
search("search_index", "key:value").first
```

Misalnya, kode resep berikut menggunakan indeks `aws_opsworks_app` pencarian untuk menemukan item tas data yang berisi nama pendek aplikasi`mylinuxdemoapp`. Kemudian menggunakan konten item tas data untuk menulis pesan ke log Chef dengan nama pendek aplikasi yang sesuai dan URL sumber:

```
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']}' **********")
```

Hanya untuk indeks `aws_opsworks_instance` pencarian, Anda dapat menentukan `self:true` untuk mewakili contoh tempat resep sedang dijalankan. Kode resep berikut menggunakan konten item tas data yang sesuai untuk menulis pesan ke log Chef dengan ID dan sistem AWS OpsWorks operasi Stacks-generated instance yang sesuai:

```
instance = search("aws_opsworks_instance", "self:true").first
Chef::Log.info("********** For instance '#{instance['instance_id']}', the instance's
 operating system is '#{instance['os']}' **********")
```

Alih-alih menggunakan pencarian Chef untuk mengakses tas data, item tas data, dan konten tas data, Anda dapat mengaksesnya secara langsung. Untuk melakukan ini, gunakan metode data_bag dan data_bag_item untuk mengakses kantong data dan item tas data, masing-masing. Misalnya, kode resep berikut melakukan hal yang sama seperti contoh sebelumnya, kecuali bahwa ia secara langsung mengakses satu item tas data dan kemudian beberapa item tas data ketika ada lebih dari satu:

```
# 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
```

Dari dua pendekatan ini, kami sarankan Anda menggunakan pencarian Chef. Semua contoh terkait dalam panduan ini menunjukkan pendekatan ini.

Topik

- Tas Data Aplikasi (aws_opsworks_app)
- Tas Data Perintah (aws_opsworks_command)
- Tas Data Cluster Amazon ECS (aws_opsworks_ecs_cluster)
- Tas Data Elastic Load Balancing (aws_opsworks_elastic_load_balancer)
- Tas Data Instance (aws_opsworks_instance)
- Tas Data Lapisan (aws_opsworks_layer)
- Tas Data Amazon RDS (aws_opsworks_rds_db_instance)
- Tas Data Tumpukan (aws_opsworks_stack)
- Tas Data Pengguna (aws_opsworks_user)

Tas Data Aplikasi (aws_opsworks_app)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Untuk event Deploy atau perintah stack Execute Recipes, mewakili setelan aplikasi.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan nama pendek dan sumber URLs aplikasi:

```
app = search("aws_opsworks_app").first
Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}' **********")

search("aws_opsworks_app").each do |app|
  Chef::Log.info("********** The app's short name is '#{app['shortname']}' **********")
  Chef::Log.info("********** The app's URL is '#{app['app_source']['url']}'
 **********")
end
```

app_id	app_source	data_sources
deploy	atribut	wilayah
aktifkan_ssl	lingkungan	name
shortname	ssl_configuration	jenis

app_id

ID aplikasi (string). GUID yang mengidentifikasi aplikasi.

app_source

Kumpulan konten yang menentukan informasi yang digunakan AWS OpsWorks Stacks untuk menyebarkan aplikasi dari repositori kontrol sumbernya. Konten bervariasi tergantung pada jenis repositori.

password

Kata sandi untuk repositori pribadi, dan `"null"` untuk repositori publik (string). Untuk bucket S3 pribadi, konten ini diatur ke kunci rahasia.

revisi

Jika repositori memiliki beberapa cabang, konten akan menentukan cabang atau versi aplikasi, seperti `"version1"` (string). Jika belum, atur statusnya ke `"null"`.

ssh_key

Sebuah kunci SSH deploy untuk mengakses repositori Git pribadi, dan `"null"` untuk repositori publik (string).

jenis

Lokasi sumber aplikasi (string). Nilai yang valid meliputi:

- `"archive"`
- `"git"`
- `"other"`
- `"s3"`

url

Di mana sumber aplikasi berada (string).

user

Nama pengguna untuk repositori pribadi, dan `"null"` untuk repositori publik (string). Untuk bucket S3 pribadi, konten diatur ke kunci akses.

atribut

Kumpulan konten yang menjelaskan struktur direktori dan konten aplikasi.

document_root

Direktori root dari pohon dokumen. Mendefinisikan jalur ke root dokumen—atau lokasi halaman beranda aplikasi, seperti `home_html` —yang relatif terhadap direktori penerapan Anda. Kecuali atribut ini ditentukan, document_root default ke. `public` Nilai `document_root` dapat dimulai hanya dengan`a-z`,, `A-Z0-9`, `_` (garis bawah) atau `-` (tanda hubung) karakter.

data_sources

Informasi yang diperlukan untuk terhubung ke database aplikasi. Jika aplikasi memiliki lapisan database terlampir, AWS OpsWorks Stacks secara otomatis menetapkan nilai yang sesuai untuk konten ini.

Nilai data_sources adalah array, dan array diakses oleh offset integral, bukan dengan kunci. Misalnya, untuk mengakses sumber data pertama aplikasi, gunakan`app[:data_sources][0][:type]`.

database_name

Nama database, yang biasanya merupakan nama pendek aplikasi (string).

jenis

Jenis instance database, biasanya `"RdsDbInstance"` (string).

arn

Contoh database Amazon Resource Name (ARN) (string).

deploy

Apakah aplikasi harus diterapkan (Boolean). `true`untuk aplikasi yang harus diterapkan dalam peristiwa siklus hidup Deploy. Dalam acara siklus hidup Pengaturan, konten ini akan digunakan `true` untuk semua aplikasi. Untuk menentukan aplikasi mana yang harus diterapkan pada sebuah instance, periksa lapisan tempat instance tersebut berada.

wilayah

Daftar domain aplikasi (daftar string).

aktifkan_ssl

Apakah dukungan SSL diaktifkan (Boolean).

lingkungan

Kumpulan variabel lingkungan yang ditentukan pengguna yang telah ditentukan untuk aplikasi. Untuk informasi selengkapnya tentang cara menentukan variabel lingkungan aplikasi, lihatMenambahkan Aplikasi. Setiap nama konten diatur ke nama variabel lingkungan dan nilai yang sesuai diatur ke nilai variabel.

name

Nama aplikasi, yang digunakan untuk tujuan tampilan (string).

shortname

Nama pendek aplikasi, yang dihasilkan oleh AWS OpsWorks Stacks dari nama (string). Shortname digunakan secara internal dan oleh resep; ini digunakan sebagai nama direktori tempat file aplikasi Anda diinstal.

ssl_configuration

sertifikat

Jika Anda mengaktifkan dukungan SSL, sertifikat SSL aplikasi; jika tidak, `"null"` (string).

rantai

Jika SSL diaktifkan, konten untuk menentukan kunci otoritas sertifikat perantara atau otentikasi klien (string).

private_key

Jika Anda mengaktifkan dukungan SSL, kunci pribadi SSL aplikasi; jika tidak, `"null"` (string).

jenis

Jenis aplikasi, yang selalu diatur `"other"` untuk tumpukan Chef 12 Linux dan Chef 12.2 Windows (string).

Tas Data Perintah (aws_opsworks_command)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Merupakan pengaturan untuk perintah yang AWS OpsWorks Stacks berjalan pada satu atau beberapa instance.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan jenis perintah dan kapan mereka dikirim:

```
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_id
instance_id	sent_at	jenis

args

Argumen untuk perintah (string).

command_id

Pengidentifikasi unik acak perintah, yang ditetapkan oleh AWS OpsWorks Stacks (string).

iam_user_id

Jika perintah dibuat oleh pelanggan, Amazon Resource Name (ARN) dari pengguna yang membuat perintah (string).

instance_id

Identifier dari instance bahwa perintah dijalankan pada (string).

sent_at

Stempel waktu ketika AWS OpsWorks Stacks menjalankan perintah (string).

jenis

Jenis perintah (string). Nilai yang valid meliputi:

- `"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"

Tas Data Cluster Amazon ECS (aws_opsworks_ecs_cluster)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Merupakan pengaturan cluster Amazon ECS.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan nama cluster Amazon ECS dan Nama Sumber Daya 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

Nama Sumber Daya Amazon (ARN) (string) cluster.

ecs_cluster_name

Nama cluster (string).

Tas Data Elastic Load Balancing (aws_opsworks_elastic_load_balancer)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Merupakan pengaturan penyeimbang beban Elastic Load Balancing.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan nama penyeimbang beban Elastic Load Balancing dan nama DNS:

```
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_nama	layer_id

elastic_load_balancer_name

Nama load balancer (string).

dns_nama

Nama DNS load balancer (string).

layer_id

AWS OpsWorks Stacks ID dari layer yang diberikan load balancer (string).

Tas Data Instance (aws_opsworks_instance)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Merupakan pengaturan instance.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan nama host instance dan: 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
```

Contoh berikut menunjukkan berbagai cara menggunakan pencarian Chef untuk mencari melalui beberapa item tas data untuk menemukan item tas data yang berisi ID EC2 instans Amazon yang ditentukan. Contoh kemudian menggunakan isi item tas data untuk menulis pesan ke log Chef dengan alamat IP publik instans yang sesuai:

```
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
```

Contoh berikut menunjukkan cara menggunakan pencarian Chef `self:true` untuk menemukan item tas data yang berisi informasi yang terkait dengan instance tempat resep sedang dijalankan. Contoh kemudian menggunakan konten item tas data untuk menulis pesan ke log Chef dengan ID yang AWS OpsWorks dihasilkan Stacks instance yang sesuai dan alamat IP publik instans:

```
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	arsitektur	auto_scaling_type
availability_zone	dibuat_at	ebs_dioptimalkan

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	diri	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 (string) AMI (Amazon Machine Image) instance.

arsitektur

Arsitektur instance, yang selalu diatur ke `"x86_64"` (string).

auto_scaling_type

Jenis penskalaan instance:`null`,`timer`, atau `load` (string).

availability_zone

Availability Zone (AZ) instance, seperti `"us-west-2a"` (string).

dibuat_at

Waktu instance dibuat, menggunakan "*yyyy*-*mm*-*ddd*T*hh*:*mm*:*ss*+*hh*:*mm*" format UTC (string). Misalnya, `"2013-10-01T08:35:22+00:00"` sesuai dengan 8:35:22 pada 10 Oktober 2013, tanpa offset zona waktu. Untuk informasi selengkapnya, lihat ISO 8601.

ebs_dioptimalkan

Apakah instans dioptimalkan EBS (Boolean).

ec2_instance_id

EC2 Contoh ID (string).

elastic_ip

Alamat IP elastis; diatur ke `"null"` jika instance tidak memiliki alamat IP Elastis (string).

hostname

Nama host, seperti `"demo1"` (string).

instance_id

ID instance, yang merupakan GUID yang AWS OpsWorks dihasilkan Stacks yang secara unik mengidentifikasi instance (string).

instance_type

Jenis instance, seperti `"c1.medium"` (string).

layer_ids

Daftar lapisan instance, diidentifikasi oleh keunikannya IDs; misalnya,`307ut64c-c7e4-40cc-52f0-67d5k1f9992c`.

os

Sistem operasi instance (string). Nilai yang valid meliputi:

- `"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

Nama DNS pribadi (string).

private_ip

Alamat IP pribadi (string).

public_dns

Nama DNS publik (string).

public_ip

Alamat IP publik (string).

root_device_type

Jenis perangkat root (string). Nilai yang valid meliputi:

- "ebs
- "instance-store"

root_device_volume_id

ID volume perangkat root (string).

diri

truejika item tas data ini berisi informasi tentang contoh di mana resep sedang dijalankan; jika tidak, false (Boolean). Nilai ini hanya tersedia untuk resep, bukan melalui AWS OpsWorks Stacks API.

ssh_host_dsa_key_fingerprint

Urutan byte yang lebih pendek yang mengidentifikasi kunci publik DSA yang lebih panjang (string).

ssh_host_dsa_key_private

Kunci pribadi yang dihasilkan DSA untuk otentikasi SSH dengan instance (string).

ssh_host_dsa_key_public

Kunci publik yang dihasilkan DSA untuk otentikasi SSH dengan instance (string).

ssh_host_rsa_key_fingerprint

Urutan byte yang lebih pendek yang mengidentifikasi kunci publik RSA yang lebih panjang (string).

ssh_host_rsa_key_private

Kunci pribadi yang dihasilkan RSA untuk otentikasi SSH dengan instance (string).

ssh_host_rsa_key_public

Kunci publik yang dihasilkan RSA untuk otentikasi SSH dengan instance (string).

status

Status instance (string). Nilai yang valid meliputi:

- `"requested"`
- `"booting"`
- `"running_setup"`
- `"online"`
- `"setup_failed"`
- `"start_failed"`
- `"terminating"`
- `"terminated"`
- `"stopped"`
- `"connection_lost"`

subnet_id

ID subnet instance (string).

virtualization_type

Jenis virtualisasi instance (string).

Tas Data Lapisan (aws_opsworks_layer)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Merupakan pengaturan lapisan.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan nama dan nama pendek lapisan:

```
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
paket	shortname	jenis
volume_configurations		

ecs_cluster_arn

Jika layer memiliki cluster Amazon ECS yang ditetapkan, Amazon Resource Name (ARN) (string) klaster Amazon ECS.

dienkripsi

`true`jika volume EBS dienkripsi; jika tidak, `false` (Boolean).

layer_id

Layer ID, yang merupakan GUID yang dihasilkan oleh AWS OpsWorks Stacks dan yang secara unik mengidentifikasi layer (string).

name

Nama layer, yang digunakan untuk mewakili lapisan di konsol (string). Hal ini dapat didefinisikan pengguna dan tidak perlu unik.

paket

Daftar paket yang akan diinstal (daftar string).

shortname

Shortname layer, yang didefinisikan pengguna (string).

jenis

Tipe layer, yang selalu diatur `"custom"` untuk Chef 12 Linux dan Chef 12.2 Windows (string).

volume_configurations

Daftar konfigurasi volume Amazon EBS.

iops

Jumlah operasi I/O per detik yang dapat didukung oleh volume.

mount_point

Direktori titik pemasangan volume.

number_of_disk

Jumlah disk dalam volume.

raid_level

Tingkat konfigurasi RAID volume.

ukuran

Ukuran volume di GiB.

volume_type

Jenis volume: tujuan umum, magnetik, IOPS yang disediakan, HDD yang dioptimalkan untuk throughput, atau HDD dingin.

Tas Data Amazon RDS (aws_opsworks_rds_db_instance)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Satu set konten kantong data yang menentukan konfigurasi instans Amazon Relational Database Service (Amazon RDS) sebagai berikut:

alamat	db_instance_identifier	db_kata sandi
db_pengguna	engine	rds_db_instance_id
region		

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan alamat instans Amazon RDS dan jenis mesin database:

```
rds_db_instance = search("aws_opsworks_rds_db_instance").first
Chef::Log.info("********** The RDS instance's address is
 '#{rds_db_instance['address']}' **********")
Chef::Log.info("********** The RDS instance's database engine type is
 '#{rds_db_instance['engine']}' **********")

search("aws_opsworks_rds_db_instance").each do |rds_db_instance|
  Chef::Log.info("********** The RDS instance's address is
 '#{rds_db_instance['address']}' **********")
```

```
  Chef::Log.info("********** The RDS instance's database engine type is
 '#{rds_db_instance['engine']}' **********")
end
```

alamat

Nama DNS instance.

port

Port instance.

db_instance_identifier

ID instans.

db_kata sandi

Kata sandi utama instans.

db_pengguna

Nama pengguna master instance.

engine

Mesin database instance, sepertimysql.

rds_db_instance_id

Nama Sumber Daya Amazon (ARN) instans.

region

Wilayah AWS instans, sepertius-west-2.

Tas Data Tumpukan (aws_opsworks_stack)

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Merupakan pengaturan tumpukan ini.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk menulis pesan ke log Chef dengan nama tumpukan dan url sumber buku masak:

```
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

Nama Sumber Daya Amazon tumpukan (ARN) (string).

custom_cookbooks_source

Satu set konten yang menentukan repositori sumber buku masak kustom.

jenis

Jenis repositori (string). Nilai yang valid meliputi:

- `"archive"`
- `"git"`
- `"s3"`

url

URL repositori, seperti `"git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git"` (string).

nama pengguna

Nama pengguna untuk repositori pribadi, dan `null` untuk repositori publik (string). Untuk bucket Private Amazon Simple Storage Service (Amazon S3), konten diatur ke kunci akses.

password

Kata sandi untuk repositori pribadi, dan `null` untuk repositori publik (string). Untuk bucket S3 pribadi, konten ini diatur ke kunci rahasia.

ssh_key

Sebuah kunci SSH deploy untuk mengakses repositori Git pribadi, dan `null` untuk repositori publik (string).

revisi

Jika repositori memiliki beberapa cabang, konten akan menentukan cabang atau versi aplikasi, seperti `"version1"` (string). Jika belum, atur statusnya ke `null`.

name

Nama tumpukan (string).

region

Wilayah AWS tumpukan (string).

stack_id

GUID yang mengidentifikasi tumpukan (string).

use_custom_cookbooks

Apakah buku masak khusus diaktifkan (Boolean).

vpc_id

Jika tumpukan berjalan di VPC, ID VPC, jika tumpukan berjalan di VPC (string).

Tas Data Pengguna (aws_opsworks_user)

Important

AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Merupakan pengaturan pengguna.

Contoh berikut menunjukkan cara menggunakan pencarian Chef untuk mencari melalui satu item tas data dan kemudian beberapa item tas data untuk menulis pesan ke log Chef dengan nama pengguna dan Nama Sumber Daya 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_id	remote_access
ssh_public_key	unix_user_id	nama pengguna

administrator_privileges

Apakah pengguna memiliki hak administrator (Boolean).

iam_user_id

Nama Sumber Daya Amazon (ARN) (string) pengguna.

remote_access

Apakah pengguna dapat menggunakan RDP untuk masuk ke instance (Boolean).

ssh_public_key

Kunci publik pengguna, seperti yang disediakan melalui konsol AWS OpsWorks Stacks atau API (string).

unix_user_id

ID Unix pengguna (nomor).

nama pengguna

Nama pengguna (string).

OpsWorks Perubahan Agen

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Agen Chef 12 Rilis

Tabel berikut menjelaskan perubahan penting pada agen Chef 12 yang dipasang AWS OpsWorks Stacks pada instans yang dikelola.

Versi Agen	Deskripsi	Tanggal Rilis
4042	• Rilis agen ini hanya berisi perubahan kecil tanpa fitur baru	7 Februari 2023
4041	• Rilis agen ini hanya berisi perubahan kecil tanpa fitur baru • Perbarui sertifikat Amazon CA	27 Januari 2023
4040	• Rilis agen ini hanya berisi perubahan kecil tanpa fitur baru	22 Juli 2022
4039	• Perbaiki integrasi ECS untuk Ubuntu AMIs	30 April 2020
4038	• Perbaiki bug saat mengirim statistik instance selama perubahan DST	5 Maret 2020

Versi Agen	Deskripsi	Tanggal Rilis
	• Hormati variabel `no_proxy` lingkungan selama pengunduhan dan pemasangan agen	
4037	• Tambahkan dukungan untuk menandatangani permintaan ke S3 URLs tanpa wilayah menggunakan SiGv4 • Hapus dukungan untuk menandatangani permintaan S3 menggunakan SigV2	4 Juni 2019
4035	• Perbaiki bug selama pengaturan ECS • Perbaiki `fstab` entri duplikat setelah perubahan jenis instance	8 Mei 2019
4033	• Tambahkan dukungan untuk Ubuntu 18.04 • Perbaiki bug instalasi agen di Amazon Linux 2	26 November 2018
4032	• Ditambahkan dukungan untuk Amazon Linux 2	24 Oktober 2018
4031	• Tambahkan dukungan untuk Amazon Linux 2018.03 • Mendukung arsip S3 publik yang dihosting di akun lain	15 Agustus 2018
4030	• Perbaiki penanganan volume untuk instance c5d	31 Mei 2018
4029	• Instal `nvme-cli` di Ubuntu 14.04 • Perbaiki pemasangan volume pada instance c5, m5 • Selalu pertahankan nama host saat restart	2 Mei 2018
4028	• Perbaiki `monit` konfigurasi untuk CentOS	20 Maret 2018
4027	• Support untuk pemasangan NVMe volume di Ubuntu 14.04 (`nvme-cli`harus diinstal secara manual) • Tidak memerlukan `name` properti untuk volume	Februari 17, 2018

Versi Agen	Deskripsi	Tanggal Rilis
4026	• Volume EBS NVMe berbasis mount menggunakan ID volume EBS • Perbaiki pemasangan volume EBS pada instans i3 • Perbaiki urutan volume EBS yang dipasang pada instance c5, m5	31 Januari 2018
4025	• Perbaiki penanganan NVMe perangkat	13 Desember 2017
4024	• Tambahkan dukungan untuk Amazon Linux 2017.09	5 Desember 2017
4023	• Tambahkan dukungan untuk integrasi CloudWatch Log	April 2, 2017
4022	• Perbarui versi klien Chef ke 12.18.31	1 Februari 2017
4021	• Tingkatkan penanganan proxy	16 Desember 2016
4020	• Perbarui versi klien Chef ke 12.16.42	8 Desember 2016
4019	• Variabel proxy sumber selama instalasi agen • Red Hat Enterprise Linux 7 sekarang menggunakan `systemd` bukan `monit` • Jangan mengatur EPEL di Red Hat Enterprise Linux 7 • Gunakan `flock` alih-alih `lockrun.c` untuk penguncian proses • Hindari keluaran ganjil `ps -p1` saat memeriksa `systemd`	19 Oktober 2016
4018	• Perbarui versi klien Chef ke 12.13.37 • Tambahkan dukungan untuk Amazon Linux 2016.09	Agustus 25, 2016

Versi Agen	Deskripsi	Tanggal Rilis
4017	• Perbarui versi klien Chef ke 12.12.15	Agustus 10, 2016
4016	• Perbaiki penghapusan instalasi agen pada sistem yang `monit` tidak digunakan	23 Juni 2016
4015	• Perbaiki pengaturan ECS untuk Amazon Linux 2016.03	17 Juni 2016
4011	• Perbarui versi klien Chef ke 12.10.24 • Tingkatkan penanganan unggahan log	Mei 19, 2016
4008	• Tambahkan dukungan untuk Amazon Linux 2016.03 • Tambahkan batas waktu untuk menginstal bundel • Tambahkan xfs to /etc/filesystems jika ada	16 Maret 2016
4007	• Perbarui versi klien Chef ke 12.7.2 • Perbaikan untuk penanganan kesalahan untuk instans lokal (server yang dihosting di luar AWS) • Tingkatkan kompatibilitas dengan koki gula terbaru • Coba lagi unduhan arsip untuk penyebaran	Maret 4, 2016
4006	• Perbarui versi klien Chef ke 12.6.0 • Jangan instal paket libxml2- devel/libxml2-dev and libxslt-devel/libxslt -dev pada instalasi agen	21 Januari 2016
4005	• Perbaiki impor ec2 dengan selalu mengaktifkan data ec2 di ohai untuk infrastruktur ec2	17 Desember 2015
4004	• AWS OpsWorks Dukungan tumpukan untuk Chef 12 Linux- Chef Client 12.5.1	3 Desember 2015

Rilis Agen Chef 11.10

Tabel berikut menjelaskan perubahan penting pada agen Chef 11.10 yang dipasang AWS OpsWorks Stacks pada instans yang dikelola.

Versi Agen	Deskripsi	Tanggal Rilis
3456	• Rilis agen ini hanya berisi perubahan kecil tanpa fitur baru • Perbarui sertifikat Amazon CA	27 Januari 2023
3455	• Rilis agen ini hanya berisi perubahan kecil tanpa fitur baru	1 November 2022
3454	• Perbaiki integrasi ECS untuk Ubuntu AMIs	28 April 2020
3453	• Perbaiki bug saat mengirim statistik instance selama perubahan DST • Perbaiki bug paket yang hilang dalam RHEL7 pengaturan • Hormati variabel `no_proxy` lingkungan selama pengunduhan dan pemasangan agen	5 Maret 2020
3452	• Jangan sertakan wilayah di URL jalur virtual Amazon S3 jika ya `us-east-1` • Ekstrak dan unggah buku masak internal ke bucket spesifik wilayah panggung • Perbaiki `fstab` entri untuk Chef 11.10 • Hapus penggunaan SigV2 untuk S3 dan dapatkan wilayah untuk bucket dalam permintaan	13 Agustus 2019
3451	• Tambahkan dukungan untuk Ruby 2.6.1	20 Maret 2019
3450	• Perbaiki atribut EBS default • Perbaiki instalasi CloudWatchLogs agen untuk Amazon Linux 2	Selasa, 03 Desember 2018

Versi Agen	Deskripsi	Tanggal Rilis
	• Perbaiki instalasi bundler untuk versi rubygem yang lebih baru dari 2.6.14 • Perbaiki dukungan arsip S3 publik	
3449	• Perbaiki penanganan volume untuk instance c5d • Perbaiki dukungan array RAID pada NVMe instance -device	5 Juni 2018
3448	• Tingkatkan Ruby versi 2.3 default ke 2.3.7 • Perbaiki pemasangan volume EBS pada instance NVMe berbasis pada instance Ubuntu 14.04 • Mendukung arsip Amazon S3 publik yang dihosting di akun lain • Perbaiki masalah `opsworks-agent` boot pada instance Red Hat Enterprise Linux	8 Mei 2018
3447	• Volume EBS NVMe berbasis mount menggunakan ID volume EBS • Perbaiki pemasangan volume EBS pada instans i3 • Perbaiki urutan volume EBS yang dipasang pada c5, m5 • Perbarui Ruby versi 2.3 default ke 2.3.6	31 Januari 2018
3446	• Perbaiki penanganan NVMe perangkat • Perbarui Ruby versi 2.3 default ke 2.3.5	14 Desember 2017
3445	• Tambahkan dukungan untuk Amazon Linux 2017.09 • Perbarui Ruby versi 2.2 default ke 2.2.8	31 Oktober 2017
3444	• Tambahkan dukungan untuk CloudWatch Log	April 1, 2017
3443	• Tingkatkan penanganan proxy	15 Desember 2016

Versi Agen	Deskripsi	Tanggal Rilis
3442	• Perbarui Ruby versi 2.3 default ke 2.3.3 • Perbarui Ruby versi 2.2 default ke 2.2.6	6 Desember 2016
3441	• Variabel proxy sumber selama instalasi agen	Oktober 21, 2016
3440	• Tambahkan dukungan untuk Amazon Linux 2016.09	September 13, 2016
3439	• Perubahan kecil; tidak ada fitur baru	29 Juli 2016
3438	• Tambahkan dukungan untuk Ruby 2.3.1 • Tingkatkan pendaftaran instans dengan kredensil dari profil instans IAM • Hapus `s3curl.pl` sisa makanan • Perbaiki pengaturan ECS untuk Amazon Linux 2016.03	17 Juni 2016
3437	• Perbarui Ruby versi 2.2 default ke 2.2.5	4 Mei 2016
3436	• Perbarui URL EPEL untuk Red Hat Enterprise Linux. PENTING: tanpa perubahan ini, instance Red Hat Enterprise Linux gagal untuk boot.	18 April 2016
3435	• Perbarui Ruby versi 2.1 default ke 2.1.9 • Meningkatkan penanganan Amazon S3 dan penerapan arsip	April 6, 2016
3434	• Tambahkan dukungan untuk Amazon Linux 2016.03 • Coba lagi instalasi paket	16 Maret 2016

Versi Agen	Deskripsi	Tanggal Rilis
3433	• Beberapa perbaikan untuk instance lokal (server yang dihosting di luar) AWS • Tingkatkan kompatibilitas dengan yang terbaru `chef-sugar` • Coba lagi unduhan arsip untuk penyebaran • Perbaiki URL instalasi permata Ruby	Februari 27, 2016
3432	• Meningkatkan penanganan karakter khusus dalam nama bucket • Perbarui `s3_file` ke versi 2.6.6 • Lewati pemasangan volume tanpa titik pemasangan yang ditentukan • Selalu restart `unicorn` alih-alih berhenti dan mulai untuk mencegah downtime selama penerapan • Selalu perbarui buku masak khusus untuk perintah `setup` • Setelah membuat pembaruan array RAID `initramfs` untuk mencegah masalah pemetaan perangkat saat reboot	20 Januari 2016
3431	• Masalah instalasi tetap `passenger` dan `unicorn` permata di lapisan Rails • Memperbarui versi default 2.0, 2.1 dan 2.2 Ruby ke 2.0.0p648, 2.1.8 dan 2.2.4 • Izinkan nama `postgres` paket diatur dalam JSON kustom • Perbarui versi default Node.js ke 0.12.9	Desember 22, 2015
3430	• Perubahan kecil; tidak ada fitur baru	November 25, 2015

Versi Agen	Deskripsi	Tanggal Rilis
3429	• Tingkatkan daemonisasi OpsWorks agen (tutup stdout/stderr) • Meningkatkan kekokohan `s3_file` sumber daya (percobaan ulang, pengecualian tertangkap)	18 November 2015
3428	• Menambahkan deteksi `postgres` adaptor berdasarkan Gemfile, memperbaiki https://github.com/aws/ opsworks-cookbooks/issues/136	17 Juni 2016
3427	• Memperbaiki masalah dengan mengambil kredensil di agen • Memperbarui versi default 2.0, 2.1 dan 2.2 Ruby ke 2.0.0p647, 2.1.7 dan 2.2.3	September 11, 2015
3426	• Diperbarui `aws-sdk` ke 1.65.0 • Meningkatkan unduhan dari Amazon S3 dengan mengganti dengan `s3curl` `s3_file cookbook` • Ubah versi Node.js default menjadi 0.12.7 • Logging ditambahkan untuk aplikasi Node.js. STDOUT dan STDERR dicatat dan diputar di direktori `shared/log` • Buat pembaruan checkout submodul buku masak khusus eksplisit • Menambahkan solusi untuk https://github.com/aws/ opsworks-cookbooks/issues/213 yang akan memeriksa untuk memastikan pemasangan bind telah dibuat sebelum direktori dibuat `deploy`	27 Agustus 2015
3425	• Dukungan ECS untuk Amazon Linux dan Ubuntu	27 Juli 2015
3424	• Perubahan kecil; tidak ada fitur baru	9 Juli 2015

Versi Agen	Deskripsi	Tanggal Rilis
3422	• Dukungan penuh dari Red Hat Enterprise Linux 7 • Jadikan `/etc/hosts` generasi lebih tahan terhadap kesalahan	29 Juni 2015
3421	• Opsi untuk mengganti nama paket database untuk Red Hat Enterprise Linux 7 • Memperbarui `monit systemd` konfigurasi untuk `systemd` mencegah pengiriman `kill` sinyal ke proses yang dipantau oleh `monit`	11 Juni 2015

AWS OpsWorks Sumber Daya Tumpukan

> **Important**
>
> AWS OpsWorks Stacks Layanan ini mencapai akhir masa pakai pada 26 Mei 2024 dan telah dinonaktifkan untuk pelanggan baru dan yang sudah ada. Kami sangat menyarankan pelanggan untuk memindahkan beban kerja mereka ke solusi lain sesegera mungkin. Jika Anda memiliki pertanyaan tentang migrasi, hubungi AWS Dukungan Tim di AWS re:Post atau melalui AWS Dukungan Premium.

Sumber daya terkait berikut dapat membantu Anda ketika bekerja dengan layanan ini.

Panduan Referensi, Alat, dan Sumber Daya Dukungan

Beberapa panduan bermanfaat, forum, info kontak, dan sumber daya lainnya tersedia dari AWS OpsWorks Stacks dan Amazon Web Services.

- AWS OpsWorks Referensi API Stacks — Deskripsi, sintaks, dan contoh penggunaan tentang tindakan AWS OpsWorks Stacks dan tipe data, termasuk parameter umum dan kode kesalahan.
- AWS OpsWorks Stacks Technical FAQ - Pertanyaan teratas yang diajukan pengembang tentang produk ini.
- AWS OpsWorks Catatan Rilis Stacks — Tinjauan tingkat tinggi dari rilis saat ini. Dokumen ini secara khusus mencatat fitur baru, koreksi, dan masalah yang diketahui.
- AWS Tools for PowerShell — Satu set PowerShell cmdlet Windows yang mengekspos fungsionalitas AWS SDK untuk .NET di lingkungan. PowerShell
- AWS Command Line Interface — Sintaks baris perintah yang seragam untuk mengakses layanan AWS. AWS CLI menggunakan satu proses penyiapan untuk mengaktifkan akses ke semua layanan yang didukung.
- AWS OpsWorks Stacks Command Line Reference - Perintah AWS OpsWorks khusus tumpukan untuk digunakan pada prompt baris perintah.

- Kelas & Lokakarya - Tautan ke kursus berbasis peran dan khusus, selain laboratorium mandiri untuk membantu mempertajam keterampilan Anda AWS dan mendapatkan pengalaman praktis.

- AWS Pusat Pengembang — Jelajahi tutorial, unduh alat, dan pelajari tentang acara AWS pengembang.
- AWS Alat Pengembang — Tautan ke alat pengembang SDKs, toolkit IDE, dan alat baris perintah untuk mengembangkan dan mengelola AWS aplikasi.
- Memulai Pusat Sumber Daya — Pelajari cara menyiapkan Akun AWS, bergabung dengan AWS komunitas, dan meluncurkan aplikasi pertama Anda.
- Hands-On Tutorial - Ikuti step-by-step tutorial untuk meluncurkan aplikasi pertama Anda. AWS
- AWS Whitepaper — Tautan ke daftar lengkap AWS whitepaper teknis, yang mencakup topik-topik seperti arsitektur, keamanan, dan ekonomi dan ditulis oleh AWS Solutions Architects atau pakar teknis lainnya.
- AWS Dukungan Pusat — Hub untuk membuat dan mengelola AWS Dukungan kasus Anda. Juga termasuk tautan ke sumber daya bermanfaat lainnya, seperti forum, teknis FAQs, status kesehatan layanan, dan AWS Trusted Advisor.
- Dukungan— Halaman web utama untuk informasi tentang Dukungan, saluran dukungan respons cepat untuk membantu Anda membangun dan menjalankan aplikasi di cloud. one-on-one
- Hubungi Kami – Titik kontak pusat untuk pertanyaan tentang tandaihan AWS , akun, peristiwa, penyalahgunaan, dan masalah lainnya.
- AWS Ketentuan Situs — Informasi terperinci tentang hak cipta dan merek dagang kami; akun, lisensi, dan akses situs Anda; dan topik lainnya.

AWS Kit Pengembangan Perangkat Lunak

Amazon Web Services menyediakan kit pengembangan perangkat lunak untuk mengakses AWS OpsWorks Stacks dari beberapa bahasa pemrograman yang berbeda. Pustaka SDK mengotomatiskan sejumlah tugas umum, termasuk menandatangani permintaan layanan Anda secara kriptografis, mencoba ulang permintaan, atau menangani respons kesalahan.

- AWS SDK untuk Java— Pengaturan dan dokumentasi lainnya
- AWS SDK untuk .NET— Pengaturan dan dokumentasi lainnya.
- AWS SDK for PHP — Dokumentasi
- AWS SDK untuk Ruby- Dokumentasi
- dokumentasi lainnya
- AWS SDK for Python (Boto)— Pengaturan dan dokumentasi lainnya

Perangkat Lunak Sumber Terbuka

AWS OpsWorks Stacks mencakup berbagai paket perangkat lunak sumber terbuka, yang diatur oleh lisensi masing-masing. Untuk informasi selengkapnya, lihat berikut ini:

- Untuk instance Chef 12 Linux, buka `THIRD_PARTY_LICENSES` file di `/opt/aws/opsworks/current` direktori pada instance.
- Untuk Chef 11.10 dan versi sebelumnya untuk Linux, unduh PDF Dokumen Atribusi Agen OpsWorks Linux.

AWS OpsWorks sejarah dokumen

Perubahan	Deskripsi	Tanggal
Pembaruan untuk AWS OpsWorks Stacks	Anda sekarang dapat menggunakan alat Lepaskan di Tempat untuk melepaskan OpsWorks instance Anda dari layanan OpsWorks Stacks, lihat Menggunakan alat AWS OpsWorks Stacks Lepaskan di Tempat dalam panduan ini.	April 11, 2024
Pembaruan untuk AWS OpsWorks Stacks	Sekarang Anda dapat memigrasikan AWS OpsWorks Tumpukan ke Manajer AWS Systems Manager Aplikasi dengan menggunakan skrip migrasi. Untuk informasi selengkapnya, lihat Memigrasi AWS OpsWorks Stacks aplikasi Anda ke Manajer AWS Systems Manager Aplikasi dalam panduan ini.	22 Desember 2022
Update untuk AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise	Prosedur pemecahan masalah sekarang tersedia yang menjelaskan apa yang dapat Anda lakukan jika pemelihar aan sistem gagal untuk server Anda AWS OpsWorks for Chef Automate atau OpsWorks untuk Puppet Enterprise. Untuk informasi selengkap nya, lihat Pemeliharaan sistem gagal untuk server Chef	29 September 2022

	Automate atau Pemeliharaan sistem gagal untuk server Puppet Enterprise. dalam panduan ini.	
Update untuk AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise	Prosedur pemecahan masalah sekarang tersedia jika server Anda AWS OpsWorks for Chef Automate atau OpsWorks untuk Puppet Enterprise memasuki status. `Connection lost` Untuk informasi selengkapnya, lihat Server Chef Automate berada dalam `Connection lost` keadaan atau server Puppet Enterprise dalam `Connection lost` status dalam panduan ini.	Maret 23, 2022
Pembaruan untuk AWS OpsWorks Stacks	Sebagai praktik terbaik keamanan, Anda sekarang dapat menambahkan kunci `aws:SourceArn` atau `aws:SourceAccount` kondisi (atau keduanya) ke kebijakan hubungan kepercayaan yang memungkinkan akses AWS OpsWorks Stacks untuk melakukan tugas di AWS layanan lain. Untuk informasi lebih lanjut, lihat Pencegahan wakil bingung lintas layanan di AWS OpsWorks Tumpukan dalam panduan ini.	4 Maret 2022

Update untuk AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise	Sebagai praktik keamanan terbaik, Anda sekarang dapat menambahkan kunci `aws:SourceArn` atau `aws:SourceAccount` kondisi (atau keduanya) ke kebijakan hubungan kepercayaan yang memungkinkan AWS OpsWorks for Chef Automate dan OpsWorks untuk akses Perusahaan Boneka untuk melakukan tugas di AWS layanan lain. Untuk informasi lebih lanjut, lihat Pencegahan wakil bingung lintas layanan dalam panduan ini.	10 Januari 2022
Update untuk AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise	AWS OpsWorks for Chef Automate dan OpsWorks untuk Puppet Enterprise telah memperbarui kebijakan yang dikelola `AWSOpsWorksCMServiceRole` dan `AWSOpsWorksCMInstanceProfileRole`, dan sekarang menyimpan rahasia di AWS Secrets Manager.	3 Mei 2021

Update untuk AWS OpsWorks for Puppet Enterprise	Versi mesin dari server OpsWorks for Puppet Enterprise yang Anda buat di konsol sekarang adalah 2019.8.5. Dengan menggunakan API, Anda dapat menentukan salah satu versi `2019` atau `2017` ketika Anda membuat server Puppet Enterprise. `DescribeServers` API sekarang mengembalikan atribut yang dipanggil `PUPPET_API_CRL` dalam hasilnya. Atribut ini berisi daftar pencabutan sertifikat untuk penggunaan internal.	28 April 2021
AWS OpsWorks Stacks menggunakan kebijakan terkelola baru	AWS OpsWorks Stacks telah mengubah kebijakan terkelola yang menyertakan izin untuk melakukan semua tindakan di AWS OpsWorks Tumpukan. Kebijakan baru adalah AWSOpsWorks_FullAccess. Untuk informasi selengkapnya tentang izin dalam kebijakan ini, lihat Contoh kebijakan.	19 Februari 2021
Migrasikan AWS OpsWorks Stacks tumpukan dari EC2 - Classic ke VPC	Dokumentasi telah ditambahkan menjelaskan cara memigrasikan AWS OpsWorks Stacks tumpukan dari EC2 - Classic ke VPC.	29 September 2020

Regenerasi starter kit untuk AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise	Dokumentasi telah ditambahkan menjelaskan cara membuat ulang starter kit untuk AWS OpsWorks for Chef Automate atau server. AWS OpsWorks for Puppet Enterprise	29 Juli 2020
AWS OpsWorks for Puppet Enterprise memungkinkan Anda membuat server yang menggunakan domain kustom, sertifikat, dan kunci pribadi	Anda sekarang dapat membuat server OpsWorks untuk Puppet Enterprise yang menggunakan domain kustom, sertifikat, dan kunci pribadi. Anda dapat memperbarui server Puppet Enterprise yang ada untuk menggunakan domain kustom dengan membuat server dari cadangan server yang ada.	17 April 2020
AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise sekarang mendukung penandaan di konsol	Anda sekarang dapat menambahkan tag ke AWS OpsWorks for Chef Automate server atau AWS OpsWorks for Puppet Enterprise master, atau ke cadangan server, dengan menggunakan salah satu AWS Management Console atau. AWS CLI Untuk informasi selengkapnya, lihat Bekerja dengan Tag (Koki) atau Bekerja dengan Tag (Boneka).	26 Februari 2020

AWS OpsWorks for Chef Automate menyederhanakan peningkatan server Chef Automate 1 yang ada ke Chef Automate 2	Anda dapat memutakhirkan AWS OpsWorks for Chef Automate server yang memenuhi syarat yang menjalankan Chef Automate 1 ke Chef Automate 2 dengan memilih Mulai upgrade pada halaman detail server Anda di konsol, atau dengan menjalankan tindakan `StartMaintenance` API. Untuk informasi selengkapnya, lihat Memutakhirkan AWS OpsWorks for Chef Automate Server ke Chef Automate 2.	24 Januari 2020
AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise	Bab baru tentang Keamanan di AWS OpsWorks CM (AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise) telah ditambahkan ke panduan ini.	23 Desember 2019
AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise mendukung penandaan	Anda sekarang dapat menambahkan tag ke AWS OpsWorks for Chef Automate server atau AWS OpsWorks for Puppet Enterprise master, atau ke cadangan server, dengan menggunakan file. AWS CLI AWS OpsWorks CM sekarang mendukung otorisasi berbasis tag.	18 Desember 2019

AWS OpsWorks for Chef Automate memungkinkan Anda membuat server yang menggunakan domain kustom, sertifikat, dan kunci pribadi	Anda sekarang dapat membuat server AWS OpsWorks for Chef Automate 2.0 yang menggunakan domain kustom, sertifikat, dan kunci pribadi. Anda dapat memperbarui server Chef Automate 2.0 yang ada untuk menggunakan domain khusus dengan membuat server dari cadangan server yang ada.	22 Oktober 2019
AWS OpsWorks Stacks sekarang mendukung Ruby 2.6.1	AWS OpsWorks Stacks mendukung Ruby 2.6.1 pada lapisan Server Aplikasi Rails di tumpukan Chef 11.10.	2 Mei 2019
AWS OpsWorks for Chef Automate sekarang mendukung Chef Automate 2.0	AWS OpsWorks for Chef Automate Server baru akan menjalankan Chef Automate 2.0, yang mencakup pembaruan untuk Chef InSpec, fitur baru dalam pemindaian dan pelaporan kepatuhan, dan Chef Infra.	30 April 2019
AWS OpsWorks for Chef Automate dan AWS OpsWorks for Puppet Enterprise	Anda sekarang dapat menggunakan AWS CloudFormation untuk membuat AWS OpsWorks for Chef Automate server atau server AWS OpsWorks for Puppet Enterprise master.	24 Januari 2019

AWS OpsWorks Tumpukan	AWS OpsWorks Stacks sekarang mendukung instance yang menjalankan Ubuntu 18.04 LTS di tumpukan Chef 12.	18 Desember 2018
AWS OpsWorks untuk Perusahaan Boneka	Menambahkan prosedur untuk menyiapkan koneksi berbasis SSH ke repositori kontrol yang menggunakan. CodeCommit	Selasa, 03 Desember 2018
AWS OpsWorks Tumpukan	AWS OpsWorks Stacks sekarang mendukung instance yang menjalankan Amazon Linux 2 di tumpukan Chef 12.	15 November 2018
AWS OpsWorks Tumpukan	AWS OpsWorks Stacks sekarang mendukung instance yang menjalankan Amazon Linux 2018.03 di tumpukan Chef 11.10.	23 Oktober 2018
AWS OpsWorks Tumpukan	AWS OpsWorks Stacks sekarang mendukung instance yang menjalankan Amazon Linux 2018.03 di tumpukan Chef 12.	23 Agustus 2018
AWS OpsWorks for Chef Automate dan OpsWorks untuk Perusahaan Boneka	OpsWorks untuk Puppet Enterprise telah ditingkat kan ke PE 2018.1.2. AWS OpsWorks for Chef Automate telah ditingkatkan ke Chef Automate 1.8.68.	29 Juni 2018

- AWS OpsWorks for Chef Automate dan OpsWorks untuk versi API Puppet Enterprise: 2016-11-01
- AWS OpsWorks Versi Stacks API: 2016-03-08

- Update dokumentasi terbaru: 2024-04-11

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna AWS OpsWorks sebelum Juni 2018.

Deskripsi	Tanggal
AWS OpsWorks Versi Stacks Chef untuk tumpukan berbasis Windows ditingkatkan ke 12.22; Versi Ruby sekarang 2.3.6.	19 April 2018
Prosedur baru untuk membuat AWS OpsWorks for Chef Automate server atau master OpsWorks untuk Puppet Enterprise dengan menggunakan. AWS CLI	Maret 23, 2018
Versi Chef Automate diperbarui menjadi 1.8; Pengaturan Kepatuhan Chef disederhanakan dengan penambahan `opsworks-audit` buku masak.	Maret 5, 2018
Menambahkan dukungan untuk acara AWS OpsWorks Stacks di Amazon CloudWatch Events.	20 Februari 2018
Menambahkan dukungan untuk jenis volume EBS baru di AWS OpsWorks Stacks, dan API baru,. DescribeOperatingSystems	25 Januari 2018
OpsWorks untuk Puppet Enterprise dan AWS OpsWorks for Chef Automate sekarang mendukung pemilihan beberapa grup keamanan saat Anda membuat server.	18 Januari 2018
Menambahkan dukungan untuk AWS OpsWorks Stacks di Wilayah Eropa (Paris).	19 Desember 2017
Menambahkan dukungan untuk AWS OpsWorks for Chef Automate dan OpsWorks untuk Puppet Enterprise di enam wilayah tambahan, dan menambahkan prosedur untuk membuat cadangan AWS OpsWorks for Chef Automate dan OpsWorks untuk server Puppet Enterprise di. AWS Management Console	18 Desember 2017

Deskripsi	Tanggal
Menambahkan yang baru OpsWorks untuk layanan dan dokumentasi Puppet Enterprise.	16 November 2017
Menambahkan dukungan untuk Amazon Linux 2017.09 ke AWS OpsWorks Stacks.	7 November 2017
Menambahkan dukungan untuk Kepatuhan Koki ke AWS OpsWorks for Chef Automate.	25 Oktober 2017
Menambahkan dukungan untuk Amazon Linux 2017.09 ke. AWS OpsWorks for Chef Automate	Oktober 9, 2017
Menambahkan topik Pemeliharaan Sistem ke AWS OpsWorks for Chef Automate Bab.	28 Juli 2017
Ditambahkan dukungan untuk tag di AWS OpsWorks Stacks.	6 Juni 2017
Ditambahkan integrasi dengan CloudWatch Log.	10 April 2017
Ditambahkan AWS OpsWorks for Chef Automate layanan baru dan dokumentasi.	1 Desember 2016
Menambahkan dukungan untuk titik akhir regional Wilayah Timur AS (Ohio).	12 Oktober 2016
Menambahkan dukungan untuk tumpukan dan instance yang menjalankan sistem operasi Amazon Linux 2016.09.	30 September 2016
Menambahkan dukungan untuk Wilayah Asia Pasifik (Seoul) dan sembilan titik akhir regional tambahan.	Agustus 15, 2016
Menambahkan dukungan untuk Node.js 0.12.15 dan Ruby 2.3 di lapisan bawaan.	6 Juli 2016
Menambahkan dukungan untuk Wilayah Asia Pasifik (Mumbai).	28 Juni 2016
Menambahkan dukungan untuk tumpukan dan instance yang menjalankan sistem operasi CentOS 7.	22 Juni 2016

Deskripsi	Tanggal
Menambahkan panduan yang menjelaskan CodePipeline dan AWS OpsWorks integrasi Stacks.	Juni 2, 2016
Menambahkan dukungan untuk tumpukan dan instance yang menjalankan sistem operasi Ubuntu 16.04 LTS.	1 Juni 2016
Menambahkan dukungan Chef 12 Linux dan dokumentasi terkait.	3 Desember 2015
Menambahkan panduan Node.js untuk Memulai.	14 Juli 2015
Menambahkan dua contoh buku masak baru ke Cookbooks 101.	14 Juli 2015
Ditambahkan dukungan untuk manajemen versi agen.	Juni 23, 2015
Ditambahkan dukungan untuk mengelola versi agen.	24 Juni 2015
Menambahkan dukungan untuk Windows kustom AMIs.	22 Juni 2015
Menambahkan tiga topik Praktik Terbaik baru.	11 Juni 2015
Menambahkan dukungan untuk tumpukan Windows.	18 Mei 2015
Menambahkan chapter Praktik Terbaik.	15 Desember 2014
Menambahkan dukungan untuk pengurasan koneksi Elastic Load Balancing dan batas waktu Shutdown khusus.	15 Desember 2014
Menambahkan dukungan untuk mendaftarkan instans yang dibuat di luar AWS OpsWorks Stacks.	9 Desember 2014
Menambahkan dukungan untuk Amazon SWF.	4 September 2014
Menambahkan dukungan untuk mengaitkan variabel lingkungan dengan aplikasi dan Cookbooks 101 yang diperluas.	16 Juli 2014
Menambahkan Cookbooks 101, pengantar tutorial untuk mengimple mentasikan buku masak.	16 Juli 2014
Menambahkan dukungan untuk CloudTrail.	Juni 4, 2014

Deskripsi	Tanggal
Menambahkan dukungan untuk Amazon RDS.	Mei 14, 2014
Menambahkan dukungan untuk Chef 11.10 dan Berkshelf.	27 Maret 2014
Menambahkan dukungan untuk volume Amazon EBS PIOPS.	16 Desember 2013
Menambahkan izin berbasis sumber daya.	5 Desember 2013
Menambahkan manajemen sumber daya.	7 Oktober 2013
Menambahkan dukungan untuk VPCs.	29 Agustus 2013
Ditambahkan dukungan untuk kustom AMIs dan Chef 11.4.	Juli 24, 2013
Menambahkan dukungan konsol untuk beberapa lapisan per instance.	1 Juli 2013
Menambahkan dukungan untuk instans yang didukung Amazon EBS, Elastic Load Balancing, dan pemantauan Amazon. CloudWatch	14 Mei 2013
Rilis awal Panduan Pengguna AWS OpsWorks Stacks.	Februari 18, 2013

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.