aws

Opsi Penerapan Aplikasi AMS Lanjutan

Panduan Pengembang Aplikasi AMS Advanced

Versi September 13, 2024

Panduan Pengembang Aplikasi AMS Advanced: Opsi Penerapan Aplikasi AMS Lanjutan

Table of Contents

Orientasi aplikasi

Selamat datang di rencana operasi AMS AWS Managed Services (AMS). Tujuan dari dokumen ini adalah untuk menjelaskan berbagai metode yang dapat Anda gunakan saat mengarahkan aplikasi Anda ke AMS setelah jaringan awal dan manajemen akses telah disiapkan, dan masalah yang harus Anda pertimbangkan ketika memilih metode tersebut.

Dokumen ini ditujukan untuk integrator sistem dan pengembang aplikasi untuk membantu dalam menentukan dan menyusun proses aplikasi untuk pelanggan AMS baru.

Apa itu orientasi aplikasi?

Orientasi aplikasi AMS mengacu pada penyebaran sumber daya dan aplikasi, sesuai kebutuhan, ke dalam infrastruktur AMS Anda. Arsitektur aplikasi dan infrastruktur pada platform AMS sangat mirip dengan melakukannya di native AWS. Mengikuti praktik terbaik desain AWS aplikasi dan infrastruktur sambil mempertimbangkan kemampuan yang disediakan oleh AMS akan menghasilkan aplikasi yang mampu dan dapat dioperasikan yang dihosting di lingkungan AMS.

Note

- AS Timur (Virginia)
- AS Barat (California Utara)
- AS Barat (Oregon)
- AS Timur (Ohio)
- Kanada (Pusat)
- Amerika Selatan (Sao Paulo)
- EU (Ireland)
- EU (Frankfurt)
- EU (London)
- Uni Eropa Barat (Paris)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)

- Asia Pasifik (Tokyo)

Wilayah Baru sering ditambahkan. Untuk mempelajari lebih lanjut, lihat Wilayah AWS dan Availability Zones.

Apa yang kita lakukan, apa yang tidak kita lakukan

AMS memberi Anda pendekatan standar untuk menerapkan infrastruktur AWS dan menyediakan manajemen operasional berkelanjutan yang diperlukan. Untuk deskripsi lengkap tentang peran, tanggung jawab, dan layanan yang didukung, lihat Deskripsi Layanan.

> **Note**
>
> Untuk meminta agar AMS menyediakan layanan AWS tambahan, ajukan permintaan layanan. Untuk informasi selengkapnya, lihat Membuat Permintaan Layanan.

- Apa yang kita lakukan:

  Setelah Anda menyelesaikan orientasi, lingkungan AMS tersedia untuk menerima permintaan perubahan (RFCs), insiden, dan permintaan layanan. Interaksi Anda dengan layanan AMS berkisar pada siklus hidup tumpukan aplikasi. Tumpukan baru diurutkan dari daftar templat yang telah dikonfigurasi sebelumnya, diluncurkan ke subnet virtual private cloud (VPC) tertentu, dimodifikasi selama masa operasionalnya melalui permintaan perubahan (RFCs), dan dipantau untuk kejadian dan insiden 24/7.

  Tumpukan aplikasi aktif dipantau dan dipelihara oleh AMS, termasuk penambalan, dan tidak memerlukan tindakan lebih lanjut selama masa pakai tumpukan kecuali perubahan diperlukan atau tumpukan dinonaktifkan. Insiden yang terdeteksi oleh AMS yang memengaruhi kesehatan dan fungsi tumpukan menghasilkan pemberitahuan dan mungkin atau mungkin tidak memerlukan tindakan Anda untuk menyelesaikan atau memverifikasi. Pertanyaan cara dan pertanyaan lainnya dapat dilakukan dengan mengirimkan permintaan layanan.

  Selain itu, AMS memungkinkan Anda mengaktifkan layanan AWS yang kompatibel yang tidak dikelola oleh AMS. Untuk informasi tentang layanan yang kompatibel dengan AWS-AMS, lihat Mode penyediaan layanan mandiri.

- Apa yang tidak kita lakukan:

  Meskipun AMS menyederhanakan penerapan aplikasi dengan menyediakan sejumlah opsi manual dan otomatis, Anda bertanggung jawab atas pengembangan, pengujian, pembaruan, dan pengelolaan aplikasi Anda. AMS menyediakan bantuan pemecahan masalah untuk masalah infrastruktur yang berdampak pada aplikasi, tetapi AMS tidak dapat mengakses atau memvalidasi konfigurasi aplikasi Anda.

Gambar Mesin AMS Amazon (AMIs)

AMS memproduksi Amazon Machine Images (AMIs) yang diperbarui setiap bulan untuk sistem operasi yang didukung AMS. Selain itu, AMS juga menghasilkan gambar yang ditingkatkan keamanan (AMIs) berdasarkan benchmark CIS Level 1 untuk subset sistem operasi yang didukung AMS. Untuk mengetahui sistem operasi mana yang memiliki gambar yang disempurnakan keamanan, lihat Panduan Pengguna Keamanan AMS, yang tersedia melalui halaman AWS Artifact -> Reports (temukan opsi Laporan di panel navigasi kiri) yang difilter untuk AWS Managed Services. Untuk mengakses AWS Artifact, dapat menghubungi CSDM Anda untuk mendapatkan petunjuk atau buka Memulai dengan AWS Artifact.

Untuk menerima peringatan saat AMS AMIs baru dirilis, Anda dapat berlangganan topik notifikasi Amazon Simple Notification Service (Amazon SNS) yang disebut “AMS AMI”. Untuk detailnya, lihat notifikasi AMS AMI dengan SNS.

Konvensi penamaan AMS AMI adalah:`customer-ams-<operating system>-<release date>-<version>`. (misalnya,`customer-ams-rhel6-2018.11-3`)

Hanya gunakan AMS AMIs yang dimulai dengan`customer`.

AMS merekomendasikan untuk selalu menggunakan AMI terbaru. Anda dapat menemukan yang terbaru AMIs dengan:

- Mencari di konsol AMS, di AMIshalaman.
- Melihat file CSV AMS AMI terbaru, tersedia dari CSDM Anda atau melalui file ZIP ini: AMS 11.2024 konten AMI dan file CSV dalam ZIP.

  Untuk file ZIP AMI sebelumnya, lihat Riwayat Dokumen.
- Menjalankan `SKMS` perintah AMS ini (diperlukan AMS SKMS SDK):

```
aws amsskms list-amis --vpc-id VPC_ID --query "Amis.sort_by(@,&Name)[?
 starts_with(Name,'customer')].[Name,AmiId,CreationTime]" --output table
```

Konten AMS AMI ditambahkan ke basis AWS AMIs, oleh sistem operasi (OS)

- Linux AMIs:
  - AWS Alat CLI
  - NTP
  - Agen Layanan Perlindungan Trend Micro Endpoint
  - Penyebaran Kode
  - PBIS/Beyond Trust AD Bridge
  - Agen SSM
  - Yum Upgrade untuk tambalan kritis
  - Skrip/perangkat lunak manajemen khusus AMS (mengontrol boot, AD join, pemantauan, keamanan, dan logging)
- Server Windows AMIs:
  - Microsoft .NET Framework 4.5
  - PowerShell 5.1
  - AWS Alat untuk Windows PowerShell
  - PowerShell Modul AMS mengontrol boot, AD join, monitoring, keamanan, dan logging
  - Agen Layanan Perlindungan Trend Micro Endpoint
  - Agen SSM
  - CloudWatch Agen
  - EC2Layanan Config (melalui Windows Server 2012 R2)
  - EC2Peluncuran (Windows Server 2016 dan Windows Server 2019)
  - EC2Launchv2 (Windows Server 2022 dan yang lebih baru)

Berbasis Linux AMIs:

- Amazon Linux 2023 (Rilis Kecil Terbaru) (AMI Minimal tidak didukung)
- Amazon Linux 2 (Rilis Kecil Terbaru)

- Amazon Linux 2 (ARM64)
- Red Hat Enterprise 7 (Rilis Kecil Terbaru)
- Red Hat Enterprise 8 (Rilis Kecil Terbaru)
- Red Hat Enterprise 9 (Rilis Kecil Terbaru)
- SUSE Linux Server Perusahaan 15 SP6
- Ubuntu Linux 18.04
- Ubuntu Linux 20.04
- Ubuntu Linux 22.04
- Ubuntu Linux 24.04
- Amazon Linux: Untuk ikhtisar produk, informasi harga, informasi penggunaan, dan informasi dukungan, lihat Amazon Linux AMI (HVM/64-bit) dan Amazon Linux 2.

  Untuk informasi selengkapnya, lihat Amazon Linux 2 FAQs.
- RedHat Enterprise Linux (RHEL): Untuk ikhtisar produk, informasi harga, informasi penggunaan, dan informasi dukungan, lihat Red Hat Enterprise Linux (RHEL) 7 (HVM).
- Ubuntu Linux 18.04: Untuk ikhtisar produk, informasi harga, informasi penggunaan, dan informasi dukungan, lihat Ubuntu 18.04 LTS - Bionic.
- SUSE Linux Enterprise Server untuk aplikasi SAP 15: SP6
  - Jalankan langkah-langkah berikut sekali per akun:
    1. Arahkan ke AWS Marketplace.
    2. Cari produk SUSE 15 SAP.
    3. Pilih Lanjutkan untuk berlangganan.
    4. Pilih Terima persyaratan.
  - Selesaikan langkah-langkah berikut setiap kali Anda perlu meluncurkan SUSE Linux Enterprise Server baru untuk aplikasi SAP 15 SP6 instance:
    1. Perhatikan ID AMI untuk SUSE Linux Enterprise Server berlangganan untuk Aplikasi SAP 15 AMI.
    2. Buat Deployment | Komponen tumpukan lanjutan | EC2 tumpukan | Buat jenis perubahan ct-14027q0sjyt1h RFC. Ganti *`InstanceAmiId`* dengan ID AWS Marketplace AMI yang Anda berlangganan.

Berbasis Windows AMIs:

Microsoft Windows Server (2016, 2019 dan 2022), berdasarkan Windows terbaru AMIs.

Untuk contoh pembuatan AMIs, lihat Membuat AMI.

AMIsAMS Offboarding:

AMS tidak membatalkan pembagian apa pun AMIs dari Anda selama offboarding untuk menghindari dampak bagi setiap depedensi Anda. Jika Anda ingin menghapus AMS AMIs dari akun Anda, Anda dapat menggunakan `cancel-image-launch-permission` API untuk menyembunyikan spesifik AMIs. Misalnya, Anda dapat menggunakan skrip di bawah ini untuk menyembunyikan semua AMS AMIs yang dibagikan dengan akun Anda sebelumnya:

```
for ami in $(aws ec2 describe-images --executable-users self --owners 027415890775 --
query 'Images[].ImageId' --output text) ;
    do
    aws ec2 cancel-image-launch-permission --image-id $ami ;
    done
```

Anda harus menginstal AWS CLI v2 agar skrip dapat dijalankan tanpa kesalahan apa pun. Untuk langkah-langkah penginstalan AWS CLI, lihat Menginstal atau memperbarui AWS CLI versi terbaru. Untuk detail tentang `cancel-image-launch-permission` perintah, lihat `cancel-image-launch-permission`.

Keamanan ditingkatkan AMIs

AMS menyediakan gambar yang disempurnakan keamanan (AMIs) berdasarkan benchmark CIS Level 1 untuk subset sistem operasi yang didukung AMS. Untuk mengetahui sistem operasi mana yang memiliki image yang disempurnakan keamanan yang tersedia, lihat Panduan Keamanan Pelanggan AWS Managed Services (AMS). Untuk mengakses panduan ini, buka AWS Artifact, pilih Laporan di panel navigasi kiri, lalu filter untuk AWS Managed Services. Untuk petunjuk tentang cara mengakses AWS Artifact, hubungi CSDM Anda atau lihat Memulai dengan AWS Artifact untuk informasi selengkapnya.

Istilah kunci AMS

- AMS Advanced: Layanan yang dijelaskan di bagian “Deskripsi Layanan” pada Dokumentasi Lanjutan AMS. Lihat Deskripsi Layanan.

- Akun Lanjutan AMS: AWS akun yang setiap saat memenuhi semua persyaratan dalam Persyaratan Orientasi Lanjutan AMS. Untuk informasi tentang manfaat AMS Advanced, studi kasus, dan untuk menghubungi staf penjualan, lihat AWS Managed Services.
- Akun Akselerasi AMS: AWS akun yang setiap saat memenuhi semua persyaratan dalam Persyaratan Akselerasi Akselerasi AMS. Lihat Memulai dengan AMS Accelerate.
- AWS Managed Services: AMS dan atau AMS Accelerate.
- Akun AWS Managed Services: Akun AMS dan atau akun AMS Accelerate.
- Rekomendasi Kritis: Rekomendasi yang dikeluarkan oleh AWS melalui permintaan layanan yang memberi tahu Anda bahwa tindakan Anda diperlukan untuk melindungi terhadap potensi risiko atau gangguan pada sumber daya Anda atau. Layanan AWS Jika Anda memutuskan untuk tidak mengikuti Rekomendasi Kritis pada tanggal yang ditentukan, Anda bertanggung jawab penuh atas segala kerugian yang diakibatkan oleh keputusan Anda.
- Konfigurasi yang Diminta Pelanggan: Perangkat lunak, layanan, atau konfigurasi lain apa pun yang tidak diidentifikasi dalam:
  - Mempercepat: Konfigurasi yang Didukung atau Akselerasi AMS; Deskripsi Layanan.
  - AMS Lanjutan: Konfigurasi yang Didukung atau AMS Lanjutan; Deskripsi Layanan.
- Komunikasi insiden: AMS mengkomunikasikan Insiden kepada Anda atau Anda meminta Insiden dengan AMS melalui Insiden yang dibuat di Support Center for AMS Accelerate dan di AMS Console for AMS. AMS Accelerate Console menyediakan ringkasan Insiden dan Permintaan Layanan di Dasbor dan tautan ke Pusat Dukungan untuk detailnya.
- Lingkungan Terkelola: Akun AMS Advanced dan atau akun AMS Accelerate yang dioperasikan oleh AMS.

  Untuk AMS Advanced, ini termasuk akun landing zone multi-akun (MALZ) dan single-account landing zone (SALZ).
- Tanggal mulai penagihan: Hari kerja berikutnya setelah AWS menerima informasi yang Anda minta di Email Onboarding AWS Managed Services. Email Onboarding AWS Managed Services mengacu pada email yang dikirim oleh Anda AWS untuk mengumpulkan informasi yang diperlukan guna mengaktifkan AWS Managed Services di akun Anda.

  Untuk akun yang Anda daftarkan selanjutnya, tanggal mulai penagihan adalah hari berikutnya setelah AWS Managed Services mengirimkan Pemberitahuan Aktivasi AWS Managed Services untuk akun yang terdaftar. Pemberitahuan Aktivasi AWS Managed Services terjadi saat:

  1. Anda memberikan akses ke AWS akun yang kompatibel dan menyerahkannya ke AWS Managed Services.

2. AWS Managed Services mendesain dan membangun Akun AWS Managed Services.

- Penghentian Layanan: Anda dapat menghentikan AWS Managed Services untuk semua akun AWS Managed Services, atau untuk akun AWS Managed Services tertentu dengan alasan apa pun dengan memberikan pemberitahuan AWS setidaknya 30 hari melalui permintaan layanan. Pada Tanggal Penghentian Layanan, baik:
  1. AWS menyerahkan kontrol semua akun AWS Managed Services atau akun AWS Managed Services yang ditentukan sebagaimana berlaku, untuk Anda, atau
  2. Para pihak menghapus AWS Identity and Access Management peran yang memberikan AWS akses dari semua akun AWS Managed Services atau akun AWS Managed Services yang ditentukan, sebagaimana berlaku.
- Tanggal penghentian layanan: Tanggal penghentian layanan adalah hari terakhir bulan kalender setelah berakhirnya periode pemberitahuan penghentian 30 hari yang diperlukan. Jika akhir periode pemberitahuan penghentian yang diperlukan jatuh setelah hari ke-20 bulan kalender, maka tanggal penghentian layanan adalah hari terakhir dari bulan kalender berikutnya. Berikut ini adalah contoh skenario untuk tanggal penghentian.
  - Jika pemberitahuan penghentian diberikan pada 12 April, maka pemberitahuan 30 hari berakhir pada 12 Mei. Tanggal penghentian layanan adalah 31 Mei.
  - Jika pemberitahuan penghentian diberikan pada 29 April, maka pemberitahuan 30 hari berakhir pada 29 Mei. Tanggal penghentian layanan adalah 30 Juni.
- Penyediaan AWS Managed Services AWS : menyediakan bagi Anda dan Anda dapat mengakses serta menggunakan AWS Managed Services untuk setiap akun AWS Managed Services sejak tanggal dimulainya layanan.
- Penghentian untuk akun AWS Managed Services tertentu: Anda dapat menghentikan AWS Managed Services untuk akun AWS Managed Services tertentu dengan alasan apa pun dengan memberikan AWS pemberitahuan melalui permintaan layanan (“Permintaan Penghentian Akun AMS”).

Ketentuan manajemen insiden:

- Acara: Perubahan di lingkungan AMS Anda.
- Peringatan: Setiap kali peristiwa dari yang didukung Layanan AWS melebihi ambang batas dan memicu alarm, peringatan dibuat dan pemberitahuan dikirim ke daftar kontak Anda. Selain itu, insiden dibuat dalam daftar Insiden Anda.

- Insiden: Gangguan yang tidak direncanakan atau penurunan kinerja lingkungan AMS atau AWS Managed Services yang menghasilkan dampak seperti yang dilaporkan oleh AWS Managed Services atau Anda.
- Masalah: Akar penyebab bersama dari satu atau lebih insiden.
- Resolusi Insiden atau Menyelesaikan Insiden:
  - AMS telah memulihkan semua layanan AMS yang tidak tersedia atau sumber daya yang berkaitan dengan insiden tersebut ke keadaan yang tersedia, atau
  - AMS telah menetapkan bahwa tumpukan atau sumber daya yang tidak tersedia tidak dapat dikembalikan ke status yang tersedia, atau
  - AMS telah memulai pemulihan infrastruktur yang diotorisasi oleh Anda.
- Waktu Respons Insiden: Perbedaan waktu antara saat Anda membuat insiden, dan saat AMS memberikan respons awal melalui konsol, email, pusat layanan, atau telepon.
- Waktu Resolusi Insiden: Perbedaan waktu antara saat AMS atau Anda membuat insiden, dan kapan insiden diselesaikan.
- Prioritas Insiden: Bagaimana insiden diprioritaskan oleh AMS, atau oleh Anda, sebagai Rendah, Sedang, atau Tinggi.
  - Rendah: Masalah non-kritis dengan layanan AMS Anda.
  - Medium: Layanan AWS dalam lingkungan terkelola Anda tersedia tetapi tidak berfungsi sebagaimana dimaksud (sesuai deskripsi layanan yang berlaku).
  - Tinggi: Baik (1) Konsol AMS, atau satu atau beberapa AMS APIs dalam lingkungan terkelola Anda tidak tersedia; atau (2) satu atau beberapa tumpukan AMS atau sumber daya dalam lingkungan terkelola Anda tidak tersedia dan ketidaktersediaannya mencegah aplikasi Anda menjalankan fungsinya.

  AMS dapat mengkategorikan ulang insiden sesuai dengan pedoman di atas.
- Pemulihan Infrastruktur: Menyebarkan kembali tumpukan yang ada, berdasarkan templat tumpukan yang terkena dampak, dan memulai pemulihan data berdasarkan titik pemulihan terakhir yang diketahui, kecuali ditentukan lain oleh Anda, ketika resolusi insiden tidak memungkinkan.

Istilah infrastruktur:

- Lingkungan produksi terkelola: Akun pelanggan tempat aplikasi produksi pelanggan berada.
- Lingkungan non-produksi yang dikelola: Akun pelanggan yang hanya berisi aplikasi non-produksi, seperti aplikasi untuk pengembangan dan pengujian.

- AMS stack: Sekelompok satu atau lebih AWS sumber daya yang dikelola oleh AMS sebagai satu unit.
- Infrastruktur yang tidak dapat diubah: Model pemeliharaan infrastruktur yang khas untuk grup Auto EC2 Scaling Amazon ASGs () di mana komponen infrastruktur yang diperbarui AWS, (dalam, AMI) diganti untuk setiap penerapan, daripada diperbarui di tempat. Keuntungan dari infrastruktur yang tidak dapat diubah adalah bahwa semua komponen tetap dalam keadaan sinkron karena selalu dihasilkan dari basis yang sama. Kekekalan tidak tergantung pada alat atau alur kerja apa pun untuk membangun AMI.
- Infrastruktur yang dapat berubah: Model pemeliharaan infrastruktur yang khas untuk tumpukan yang bukan grup Auto EC2 Scaling Amazon dan berisi satu instance atau hanya beberapa instance. Model ini paling dekat mewakili penyebaran sistem tradisional, berbasis perangkat keras, di mana sistem digunakan pada awal siklus hidupnya dan kemudian pembaruan berlapis ke sistem itu dari waktu ke waktu. Setiap pembaruan pada sistem diterapkan ke instance satu per satu, dan dapat menyebabkan downtime sistem (tergantung pada konfigurasi tumpukan) karena aplikasi atau sistem restart.
- Grup keamanan: Firewall virtual untuk instans Anda untuk mengontrol lalu lintas masuk dan keluar. Grup keamanan bertindak pada tingkat instans, bukan tingkat subnet. Oleh karena itu, setiap instance dalam subnet di VPC Anda dapat memiliki kumpulan grup keamanan yang berbeda yang ditugaskan padanya.
- Perjanjian Tingkat Layanan (SLAs): Bagian dari kontrak AMS dengan Anda yang menentukan tingkat layanan yang diharapkan.
- SLA Tidak Tersedia dan Tidak Tersedia:
  - Permintaan API yang dikirimkan oleh Anda yang menghasilkan kesalahan.
  - Permintaan Konsol yang dikirimkan oleh Anda yang menghasilkan respons HTTP 5xx (server tidak mampu melakukan permintaan).
  - Setiap Layanan AWS penawaran yang merupakan tumpukan atau sumber daya dalam infrastruktur yang dikelola AMS Anda berada dalam keadaan “Gangguan Layanan” seperti yang ditunjukkan di Dashboard Service Health.
  - Ketidaktersediaan yang dihasilkan secara langsung atau tidak langsung dari pengecualian AMS tidak dipertimbangkan dalam menentukan kelayakan untuk kredit layanan. Layanan dianggap tersedia kecuali memenuhi kriteria karena tidak tersedia.
- Tujuan Tingkat Layanan (SLOs): Bagian dari kontrak AMS dengan Anda yang menentukan sasaran layanan khusus untuk layanan AMS.

Ketentuan penambalan:

- Patch wajib: Pembaruan keamanan penting untuk mengatasi masalah yang dapat membahayakan keadaan keamanan lingkungan atau akun Anda. “Pembaruan Keamanan Kritis” adalah pembaruan keamanan yang dinilai sebagai “Kritis” oleh vendor sistem operasi yang didukung AMS.
- Patch diumumkan versus dirilis: Patch umumnya diumumkan dan dirilis sesuai jadwal. Patch yang muncul diumumkan ketika kebutuhan akan tambalan telah ditemukan dan, biasanya segera setelah itu, tambalan dilepaskan.
- Patch add-on: Patching berbasis tag untuk instans AMS yang memanfaatkan fungsionalitas AWS Systems Manager (SSM) sehingga Anda dapat menandai instance dan menambal instance tersebut menggunakan baseline dan jendela yang Anda konfigurasikan.
- Metode tambalan:
  - Penambalan di tempat: Penambalan yang dilakukan dengan mengubah instance yang ada.
  - Patching pengganti AMI: Penambalan yang dilakukan dengan mengubah parameter referensi AMI dari konfigurasi peluncuran grup Auto EC2 Scaling yang ada.
- Penyedia patch (vendor OS, pihak ketiga): Patch disediakan oleh vendor atau badan pengatur aplikasi.
- Jenis Patch:
  - Pembaruan Keamanan Kritis (CSU): Pembaruan keamanan yang dinilai sebagai “Kritis” oleh vendor sistem operasi yang didukung.
  - Pembaruan Penting (IU): Pembaruan keamanan yang dinilai sebagai “Penting” atau pembaruan non-keamanan yang dinilai sebagai “Kritis” oleh vendor sistem operasi yang didukung.
  - Other Update (OU): Pembaruan oleh vendor dari sistem operasi yang didukung yang bukan CSU atau IU.
- Patch yang didukung: AMS mendukung patch tingkat sistem operasi. Upgrade dirilis oleh vendor untuk memperbaiki kerentanan keamanan atau bug lain atau untuk meningkatkan kinerja. Untuk daftar dukungan saat ini OSs, lihat Support Configurations.

Ketentuan keamanan:

- Kontrol Detektif: Pustaka monitor yang dibuat atau diaktifkan oleh AMS yang menyediakan pengawasan berkelanjutan terhadap lingkungan dan beban kerja yang dikelola pelanggan untuk konfigurasi yang tidak selaras dengan kontrol keamanan, operasional, atau pelanggan, dan mengambil tindakan dengan memberi tahu pemilik, memodifikasi, atau menghentikan sumber daya secara proaktif.

Ketentuan Permintaan Layanan:

- Permintaan layanan: Permintaan oleh Anda untuk tindakan yang Anda ingin AMS ambil atas nama Anda.
- Pemberitahuan peringatan: Pemberitahuan yang diposting oleh AMS ke halaman daftar permintaan Layanan Anda saat peringatan AMS dipicu. Kontak yang dikonfigurasi untuk akun Anda juga diberitahukan oleh metode yang dikonfigurasi (misalnya, email). Jika Anda memiliki tag kontak pada instans/sumber daya Anda, dan telah memberikan persetujuan kepada manajer pengiriman layanan cloud (CSDM) Anda untuk pemberitahuan berbasis tag, informasi kontak (nilai kunci) dalam tag juga diberitahukan untuk peringatan AMS otomatis.
- Pemberitahuan layanan: Pemberitahuan dari AMS yang diposting ke halaman daftar permintaan Layanan Anda.

Istilah lain-lain:

- AWS Managed Services Interface: Untuk AMS: AWS Managed Services Advanced Console, AMS CM API, dan Dukungan API. Untuk AMS Accelerate: Dukungan Konsol dan Dukungan API.
- Kepuasan Pelanggan (CSAT): AMS CSAT diinformasikan dengan analitik mendalam termasuk Peringkat Korespondensi Kasus pada setiap kasus atau korespondensi saat diberikan, survei triwulanan, dan sebagainya.
- DevOps: DevOps adalah metodologi pengembangan yang sangat menganjurkan otomatisasi dan pemantauan di semua langkah. DevOps bertujuan untuk siklus pengembangan yang lebih pendek, peningkatan frekuensi penyebaran, dan rilis yang lebih dapat diandalkan dengan menyatukan fungsi pengembangan dan operasi yang terpisah secara tradisional di atas fondasi otomatisasi. Ketika pengembang dapat mengelola operasi, dan operasi menginformasikan pengembangan, masalah dan masalah lebih cepat ditemukan dan diselesaikan, dan tujuan bisnis lebih mudah dicapai.
- ITIL: Perpustakaan Infrastruktur Teknologi Informasi (disebut ITIL) adalah kerangka kerja ITSM yang dirancang untuk membakukan siklus hidup layanan TI. ITIL diatur dalam lima tahap yang mencakup siklus hidup layanan TI: strategi layanan, desain layanan, transisi layanan, operasi layanan, dan peningkatan layanan.
- Manajemen layanan TI (ITSM): Serangkaian praktik yang menyelaraskan layanan TI dengan kebutuhan bisnis Anda.
- Managed Monitoring Services (MMS): AMS mengoperasikan sistem pemantauan sendiri, Managed Monitoring Service (MMS), yang mengkonsumsi peristiwa AWS Kesehatan dan mengumpulkan

data CloudWatch Amazon, dan data dari Layanan AWS lainnya, memberi tahu operator AMS (online 24x7) dari alarm apa pun yang dibuat melalui topik Amazon Simple Notification Service (Amazon SNS).

- Namespace: Saat Anda membuat kebijakan IAM atau bekerja dengan Amazon Resource Names (ARNs), Anda mengidentifikasi Layanan AWS dengan menggunakan namespace. Anda menggunakan ruang nama saat mengidentifikasi tindakan dan sumber daya.

Apa model operasi saya?

Sebagai pelanggan AMS, organisasi Anda telah memutuskan untuk memisahkan operasi aplikasi dan infrastruktur dan menggunakan AMS untuk operasi infrastruktur. AMS akan bekerja dengan tim desain dan pengembangan aplikasi Anda bersama dengan tim desain infrastruktur Anda untuk memastikan bahwa operasi infrastruktur Anda berjalan dengan lancar. Grafik berikut menggambarkan konsep ini:

AMS bertanggung jawab atas operasi AWS infrastruktur Anda sementara tim Anda bertanggung jawab atas operasi aplikasi Anda. Sebagai tim desain aplikasi dan infrastruktur, Anda harus memahami siapa yang akan mengoperasikan aplikasi setelah digunakan untuk produksi di infrastruktur AMS. Panduan ini mencakup pendekatan umum untuk desain infrastruktur yang berkaitan dengan penerapan dan pemeliharaan aplikasi.

Manajemen layanan di AWS Managed Services

Topik

Bagaimana layanan AMS bekerja untuk Anda.

Tata kelola akun di AWS Managed Services

Bagian ini mencakup tata kelola akun AMS.

Anda ditunjuk sebagai manajer pengiriman layanan cloud (CSDM) yang memberikan bantuan konsultasi di seluruh AMS, dan memiliki pemahaman terperinci tentang kasus penggunaan dan arsitektur teknologi Anda untuk lingkungan terkelola. CSDMs bekerja dengan manajer akun, manajer akun teknis, arsitek cloud AWS Managed Services (CAs), dan arsitek solusi AWS (SAs), sebagaimana berlaku, untuk membantu meluncurkan proyek baru dan memberikan rekomendasi praktik terbaik di seluruh proses pengembangan dan operasi perangkat lunak. CSDM adalah titik kontak utama untuk AMS. Tanggung jawab utama CSDM Anda adalah:

- Atur dan pimpin pertemuan tinjauan layanan bulanan dengan pelanggan.
- Berikan detail tentang keamanan, pembaruan perangkat lunak untuk lingkungan dan peluang untuk pengoptimalan.
- Juara persyaratan Anda termasuk permintaan fitur untuk AMS.
- Menanggapi dan menyelesaikan permintaan pelaporan penagihan dan layanan.
- Memberikan wawasan untuk rekomendasi optimalisasi keuangan dan kapasitas.

Dimulainya layanan di AWS Managed Services

Dimulainya Layanan: Tanggal Dimulainya Layanan untuk akun AWS Managed Services adalah hari pertama bulan kalender pertama setelah AWS memberi tahu Anda bahwa aktivitas yang ditetapkan dalam Persyaratan Orientasi untuk akun AWS Managed Services telah selesai; dengan ketentuan bahwa jika AWS membuat pemberitahuan tersebut setelah hari ke-20 dalam satu bulan kalender, Tanggal Dimulainya Layanan adalah hari pertama bulan kalender kedua setelah tanggal dari pemberitahuan tersebut.

Dimulainya Layanan

- R adalah singkatan dari pihak yang bertanggung jawab yang melakukan pekerjaan untuk mencapai tugas.
- Saya singkatan dari Informed; sebuah pihak yang diinformasikan tentang kemajuan, seringkali hanya pada penyelesaian tugas atau deliverable.

Dimulainya layanan

Langkah #	Judul langkah	Deskripsi	Pelangga	AMS
1.	Serah terima akun AWS pelanggan	Pelanggan membuat akun AWS baru dan menyerahkannya ke AWS Managed Services	R	I
2.	Akun AWS Managed Services - desain	Menyelesaikan desain Akun AWS Managed Services	I	R
3.	Akun AWS Managed Services - build	Akun AWS Managed Services dibuat sesuai desain di Langkah 2	I	R

Manajemen hubungan pelanggan (CRM)

AWS Managed Services (AMS) menyediakan proses manajemen hubungan pelanggan (CRM) untuk memastikan bahwa hubungan yang terdefinisi dengan baik dibuat dan dipertahankan dengan Anda.

Dasar dari hubungan ini didasarkan pada wawasan AMS tentang kebutuhan bisnis Anda. Proses CRM memfasilitasi pemahaman yang akurat dan komprehensif tentang:

- Kebutuhan bisnis Anda dan cara mengisi kebutuhan tersebut
- Kemampuan dan kendala Anda
- AMS dan tanggung jawab dan kewajiban Anda yang berbeda

Proses CRM memungkinkan AMS untuk menggunakan metode yang konsisten untuk memberikan layanan kepada Anda dan menyediakan tata kelola untuk hubungan Anda dengan AMS. Proses CRM meliputi:

- Mengidentifikasi pemangku kepentingan utama Anda
- Membangun tim tata kelola
- Melakukan dan mendokumentasikan pertemuan tinjauan layanan dengan Anda
- Menyediakan prosedur pengaduan layanan formal dengan prosedur eskalasi
- Menerapkan dan memantau kepuasan dan proses umpan balik Anda
- Mengelola kontrak Anda

Proses CRM

Proses CRM mencakup kegiatan-kegiatan ini:

- Mengidentifikasi dan memahami proses dan kebutuhan bisnis Anda. Perjanjian Anda dengan AMS mengidentifikasi pemangku kepentingan Anda.
- Mendefinisikan layanan yang akan diberikan untuk memenuhi kebutuhan dan persyaratan Anda.
- Bertemu dengan Anda dalam rapat tinjauan layanan untuk membahas setiap perubahan dalam lingkup layanan AMS, SLA, kontrak, dan kebutuhan bisnis Anda. Pertemuan sementara dapat diadakan dengan Anda untuk membahas kinerja, prestasi, masalah, dan rencana tindakan.
- Memantau kepuasan Anda dengan menggunakan survei kepuasan pelanggan kami dan umpan balik yang diberikan pada rapat.
- Melaporkan kinerja pada laporan kinerja bulanan yang diukur secara internal.
- Meninjau layanan dengan Anda untuk menentukan peluang perbaikan. Ini termasuk komunikasi yang sering dengan Anda mengenai tingkat dan kualitas layanan AMS yang disediakan.

Rapat CRM

Manajer pengiriman layanan cloud AMS (CSDMs) melakukan pertemuan dengan Anda secara teratur untuk mendiskusikan jalur layanan (operasi, keamanan, dan inovasi produk) dan trek eksekutif (laporan SLA, ukuran kepuasan, dan perubahan kebutuhan bisnis Anda).

Rapat	Tujuan	Mode	Peserta
Ulasan status mingguan (opsional)	Masalah atau insiden yang luar biasa, penambalan, peristiwa keamanan, catatan masalah Tren operasional 12 minggu (+/- 6) Kekhawatiran operator aplikasi Jadwal akhir pekan	Pelanggan di tempat location/ Telecom/Chime	AMS: CSDM dan arsitek cloud (CA) Anggota tim yang ditugaskan pelanggan (mis.: Cloud/ Infrastructure, Application Support, Architecture teams, dll.)
Ulasan bisnis bulanan	Meninjau kinerja tingkat layanan (laporan, analisis, dan tren) Analisis keuangan Peta jalan produk CSAT	Pelanggan di tempat location/ Telecom/Chime	AMS: CSDM, arsitek cloud (CA), tim akun AMS, manajer produk teknis AMS (TPM) (opsional), manajer AMS OPS (opsional) Anda: Perwakilan Operator Aplikasi

Rapat	Tujuan	Mode	Peserta
Ulasan bisnis triwulanan	Kinerja dan tren Scorecard dan Service Level Agreement (SLA) (6 bulan) Rencana/migrasi 3/6/9/12 bulan mendatang Mitigasi risiko dan risiko Inisiatif perbaikan utama Item peta jalan produk Peluang selaras arah masa depan Keuangan Inisiatif penghematan biaya Optimalisasi bisnis	Lokasi pelanggan di tempat	AMS: CSDM, arsitek cloud, tim akun AMS, direktur layanan AMS, manajer operasi AMS Anda: Perwakilan operator aplikasi, perwakilan layanan, direktur layanan

Pengaturan Rapat CRM

AMS CSDM bertanggung jawab untuk mendokumentasikan pertemuan, termasuk:

- Membuat agenda, termasuk item tindakan, masalah, dan daftar peserta.
- Membuat daftar item tindakan yang ditinjau pada setiap pertemuan untuk memastikan item selesai dan diselesaikan sesuai jadwal.
- Mendistribusikan notulen rapat dan daftar item tindakan kepada peserta rapat melalui email dalam satu hari kerja setelah rapat.
- Menyimpan notulen rapat di repositori dokumen yang sesuai.

Dengan tidak adanya CSDM, perwakilan AMS yang memimpin rapat membuat dan mendistribusikan risalah.

> **Note**
>
> CSDM Anda bekerja sama dengan Anda untuk menetapkan tata kelola akun Anda.

Laporan bulanan CRM

AMS CSDM Anda mempersiapkan dan mengirimkan presentasi kinerja layanan bulanan. Presentasi mencakup informasi tentang hal-hal berikut:

- Tanggal laporan
- Ringkasan dan Wawasan:
  - Key Call Out: jumlah tumpukan total dan aktif, status tambalan tumpukan, status orientasi akun (hanya selama orientasi), ringkasan masalah khusus pelanggan
  - Performa: Statistik tentang resolusi insiden, peringatan, penambalan, permintaan perubahan (RFCs), permintaan layanan, dan ketersediaan konsol dan API
  - Masalah, tantangan, kekhawatiran, dan risiko: Status masalah khusus pelanggan
  - Item mendatang: Rencana orientasi atau resolusi insiden khusus pelanggan
- Sumber Daya Terkelola: Grafik dan diagram lingkaran tumpukan
- Metrik AMS: Metrik pemantauan dan peristiwa, metrik insiden, metrik kepatuhan AMS SLA, metrik permintaan layanan, metrik manajemen perubahan, metrik penyimpanan, metrik kontinuitas, metrik Trusted Advisor, dan ringkasan biaya (disajikan beberapa cara). Permintaan fitur. Informasi kontak.

> **Note**
>
> Selain informasi yang dijelaskan, CSDM Anda juga memberi tahu Anda tentang perubahan material dalam ruang lingkup atau ketentuan, termasuk penggunaan subkontraktor oleh AMS untuk kegiatan operasional.
> AMS menghasilkan laporan tentang penambalan dan pencadangan yang disertakan CSDM Anda dalam laporan bulanan Anda. Sebagai bagian dari sistem pembuatan laporan, AMS menambahkan beberapa infrastruktur ke akun Anda yang tidak dapat diakses oleh Anda:
>
> - Bucket S3, dengan data mentah dilaporkan
> - Contoh Athena, dengan definisi kueri untuk menanyakan data
> - Glue Crawler untuk membaca data mentah dari bucket S3

Optimalisasi biaya di AWS Managed Services

AWS Managed Services menyediakan laporan pemanfaatan biaya dan penghematan terperinci setiap bulan kepada Anda selama tinjauan bisnis bulanan Anda (MBRs).

AMS mengikuti serangkaian proses dan mekanisme standar untuk mengidentifikasi jalan penghematan biaya di akun terkelola Anda dan membantu Anda merencanakan dan meluncurkan perubahan untuk mengoptimalkan pengeluaran AWS Anda.

> **Note**
>
> AMS sedang mengembangkan video untuk membantu optimasi biaya. Langkah pertama adalah memberi Anda PDF dan spreadsheet Excel tentang praktik terbaik pengoptimalan biaya. Untuk mengakses sumber daya ini, buka Panduan cepat untuk pengoptimalan biaya file ZIP.

Kerangka kerja pengoptimalan biaya

AMS mengikuti pendekatan tiga tahap dengan Anda untuk mengoptimalkan biaya AWS Anda:

1. Identifikasi jalan pengoptimalan biaya di lingkungan terkelola Anda
2. Sajikan rencana pengoptimalan biaya untuk Anda
3. Membantu dalam mencapai optimalisasi biaya dengan cara yang terukur

Identifikasi jalan optimasi biaya di lingkungan yang dikelola

AMS menggunakan alat AWS asli seperti Cost explorer, dan Trusted Advisor sambil memanfaatkan lebih dari 20 pola penghematan biaya di seluruh pengoptimalan arsitektur EC2 , instans, AWS dan pengoptimalan yang berfokus pada akun untuk membangun rekomendasi penghematan biaya yang disesuaikan untuk Anda.

Beberapa rekomendasi pengoptimalan meliputi yang berikut ini.

Rekomendasi optimasi arsitektur:

- Penggunaan kelas penyimpanan S3 yang optimal: Amazon S3 menawarkan berbagai kelas penyimpanan untuk memenuhi berbagai persyaratan beban kerja berdasarkan akses data,

ketahanan, dan biaya. Analisis kelas penyimpanan S3 Intelligent-Tiering dan S3 berdasarkan kebutuhan beban kerja memungkinkan Anda mengelola biaya S3 secara efisien.

- Menggunakan arsitektur caching: Memanfaatkan instance cache, jika berlaku, dapat membantu Anda mengganti beberapa instance database, sekaligus memenuhi persyaratan IOPS Anda.
- Penghematan peningkatan EBS: Migrasi volume EBS Anda dari gp2 ke gp3 memberikan penghematan biaya hingga 20% dan Anda dapat memanfaatkan kinerja dasar 3.000 IOPS yang dapat diprediksi dan 125 MiB/s, terlepas dari ukuran volume.
- Menggunakan elastisitas: Kemampuan auto-scaling yang menyediakan memungkinkan pemanfaatan sumber daya AWS yang efektif dan jalan untuk optimalisasi biaya. Meninjau dan memperbarui kebijakan penskalaan instans secara teratur berdasarkan kebutuhan, selanjutnya memberikan penghematan biaya.

EC2 rekomendasi yang berfokus pada contoh

- Pengukuran instans: Rekomendasi berfokus pada ukuran instance dan konfigurasi optimal berdasarkan penggunaan. Rekomendasi juga mencakup penggunaan fitur Amazon EC2 Auto Scaling dan EC2 mengganti instans jika berlaku AWS Lambda dengan atau konten web statis di Amazon S3, dll.
- Penjadwalan instans: Menggunakan Penjadwal Sumber Daya AMS untuk memulai dan menghentikan instans secara otomatis berdasarkan jadwal waktu membantu menahan biaya, terutama untuk instans non-produksi yang tidak digunakan selama jam non-bisnis.
- Berlangganan paket Tabungan: Paket tabungan adalah cara termudah untuk menghemat AWS penggunaan. EC2 Instance Savings Plans menawarkan penghematan hingga 72% dibandingkan dengan harga Sesuai Permintaan untuk penggunaan EC2 instans Amazon Anda. Amazon SageMaker AI Savings Plans menawarkan penghematan hingga 64% untuk penggunaan layanan Amazon SageMaker AI Anda. AMS memberikan rekomendasi yang sesuai tentang paket Tabungan berdasarkan penggunaan AWS sumber daya Anda.
- Panduan penggunaan dan konsumsi instans cadangan (RI): Instans EC2 Cadangan Amazon (RI) memberikan diskon signifikan (hingga 75%) dibandingkan dengan harga Sesuai Permintaan dan menyediakan reservasi kapasitas saat digunakan di zona ketersediaan tertentu.
- Penggunaan instans spot: Beban kerja toleran kesalahan dapat memanfaatkan instans Spot dan mengurangi harga hingga 90%.
- Pengakhiran instans idle: Mengidentifikasi dan melaporkan instance yang menganggur atau memiliki pemanfaatan rendah yang dapat dihentikan.

Rekomendasi yang berfokus pada akun

- Pembersihan akun: Pada tingkat akun, AMS juga mengidentifikasi volume EBS yang tidak digunakan, CloudTrail jejak duplikat, akun kosong dengan sumber daya yang tidak digunakan, dan sebagainya, dan memberikan rekomendasi untuk pembersihan.
- Rekomendasi SLA: Selanjutnya, AMS secara teratur meninjau akun Plus dan Premium Anda dan merekomendasikan memilih tingkat SLA yang tepat untuk akun tersebut.
- Optimalisasi otomatisasi AMS: AMS terus memilih otomatisasi AMS dan infrastruktur yang digunakan untuk menyediakan layanan AMS.

Hadir untuk pelanggan dan membantu dalam perencanaan

AMS melakukan tinjauan bisnis bulanan (MBRs) dengan pemangku kepentingan pelanggan utama dan menyajikan jalan penghematan biaya, mekanisme, dan rekomendasi yang diidentifikasi bersama dengan potensi penghematan biaya. Kami selanjutnya bekerja dengan Anda untuk merencanakan perubahan yang diperlukan.

Membantu dalam implementasi rekomendasi dan mengukur dampak biaya

AMS membantu dalam mencapai dan mengukur dampak biaya dan perubahan optimasi.

Anda menilai dampak aplikasi, risiko, dan kriteria keberhasilan dari perubahan yang disarankan, dan mengajukan permintaan perubahan (RFCs) yang sesuai melalui konsol AMS. AMS berkolaborasi dengan Anda dan mengimplementasikan perubahan yang terkait dengan pengoptimalan biaya di akun terkelola Anda. AMS mengukur dampak biaya dan memasukkan penghematan yang direalisasikan dalam tinjauan bisnis bulanan (MBRs).

Matriks tanggung jawab optimasi biaya

Tanggung jawab dalam optimasi biaya AMS.

Optimalisasi biaya RACI

Aktifitas	Pelanggan	AMS
Menyusun rekomendasi penghemat	I	R

Aktifitas	Pelanggan	AMS
an biaya dan menyiapkan laporan		
Menyajikan laporan penghematan biaya	C	R
Perubahan perencanaan yang terkait dengan penghematan biaya	R	C
Menilai dampak dan risiko perubahan	R	C
Meningkatkan RFCs untuk menerapkan perubahan	R	C

Aktifitas	Pelanggan	AMS
Meninjau RFCs dan menerapkan perubahan	C	R
Menguji aplikasi dan memvalidasi implementasi perubahan	R	C
Mengukur dampak biaya pasca perubahan dan presentasi kepada pelanggan	I	R

Jam layanan di AWS Managed Services

Fitur	AMS Lanjutan
	Tingkat Premium
Permintaan layanan	24/7

Fitur	AMS Lanjutan
	Tingkat Premium
Manajemen insiden (P2-P3)	24/7
Pencadangan dan pemulihan	24/7
Manajemen tambalan	24/7
Pemantauan dan peringatan	24/7
Permintaan otomatis untuk perubahan (RFC)	24/7
Permintaan perubahan non-otomatis (RFC)	24/7
Manajer pengiriman layanan cloud (CSDM)	Senin sampai Jumat: 08:00 — 17:00, jam kerja lokal

Mendapatkan bantuan di AWS Managed Services

AMS mendukung Anda dengan Manajemen Insiden, Manajemen Permintaan Layanan, dan Manajemen Perubahan 24 jam sehari, 7 hari seminggu, 365 hari setahun (sesuai dengan Perjanjian Tingkat Layanan AMS yang diterapkan pada akun).

Untuk melaporkan masalah kinerja layanan AWS atau AMS yang memengaruhi lingkungan terkelola Anda, gunakan konsol AMS dan kirimkan laporan insiden. Untuk detailnya, lihat Melaporkan insiden. Untuk informasi umum tentang manajemen insiden AMS, lihat Respons insiden.

Untuk meminta informasi atau saran, atau meminta layanan tambahan dari AMS, gunakan konsol AMS dan kirimkan permintaan layanan. Untuk detailnya, Membuat Permintaan Layanan. Untuk informasi umum tentang permintaan layanan AMS, lihat Manajemen Permintaan Layanan.

Pengembangan aplikasi

Proses dan praktik pengembangan aplikasi yang memungkinkan desain dan penerapan aplikasi yang efektif ke dalam lingkungan AWS Managed Services (AMS). AMS memandu Anda melalui proses tingkat tinggi berikut:

1. Bayangkan dan arsitek aplikasi yang akan dikembangkan atau diintegrasikan ke lingkungan yang dikelola AMS Anda. Beberapa pertimbangan:
    a. Bagaimana Anda akan menyebarkan aplikasi Anda? Dengan otomatisasi menggunakan alat penyebaran seperti Ansible, atau secara manual dengan langsung mengunggah file yang diperlukan?
    b. Bagaimana Anda akan memperbarui aplikasi Anda? Dengan pendekatan yang dapat berubah yang memperbarui setiap instance secara terpisah, atau dengan pendekatan yang tidak dapat diubah memperbarui setiap instance dengan satu AMI yang diperbarui dalam grup Auto Scaling?
2. Rencanakan dan arsitek infrastruktur yang akan digunakan untuk meng-host aplikasi menggunakan perpustakaan AWS arsitektur, panduan AWS “Well-Architected”, dan AMS dan pakar materi pelajaran arsitektur cloud lainnya. Bagian berikut dari panduan ini memberikan informasi yang dapat membantu dalam hal ini.
3. Pilih pendekatan penyebaran infrastruktur:
    a. Full Stack: Semua komponen infrastruktur dikerahkan sekaligus, bersama-sama.
    b. Tier dan Tie: Penerapan infrastruktur dikerahkan secara terpisah dan, setelah itu, diikat bersama dengan modifikasi grup keamanan. Jenis penyebaran ini juga dicapai dengan konfigurasi serial komponen tumpukan yang dibangun di atas satu sama lain; misalnya, menentukan penyeimbang beban yang sebelumnya Anda buat saat membuat grup Auto Scaling.
    c. Lingkungan apa, seperti Dev, Staging, dan Prod, yang akan Anda gunakan?
4. Pilih tipe perubahan AMS (CTs) yang akan menyediakan tumpukan, atau tingkatan yang diperlukan, dan siapkan permintaan perubahan () RFCs yang diperlukan.
5. Kirim RFCs untuk memicu penyebaran infrastruktur ke lingkungan yang sesuai.
6. Menyebarkan aplikasi menggunakan pendekatan penerapan aplikasi yang dipilih.
7. Kerjakan ulang infrastruktur dan aplikasi sesuai kebutuhan.

8. Terapkan infrastruktur dan aplikasi ke lingkungan tindak lanjut yang sesuai, dengan asumsi penerapan pertama Anda adalah ke lingkungan non-produksi.
9. Pemeliharaan berkelanjutan ditangani oleh AMS yang mengoperasikan infrastruktur yang mendasarinya, dan tim operasi Anda yang mengoperasikan infrastruktur aplikasi.
10. Untuk menonaktifkan aplikasi, hentikan infrastruktur AMS untuknya.

Menjadi arsitek dengan baik

AWS Kami percaya bahwa sistem yang dirancang dengan baik sangat meningkatkan kemungkinan keberhasilan bisnis. Pusat AWS Arsitektur memberikan panduan ahli tentang arsitektur di. AWS Cloud

Kami merekomendasikan artikel dan whitepaper berikut untuk membantu Anda memahami pro dan kontra dari keputusan yang harus Anda buat saat membangun sistem. AWS

Apakah Anda Well-Architected? : Memperkenalkan Kerangka AWS Well-Architected, berdasarkan sekitar enam pilar:

- Keunggulan operasional: Pilar keunggulan operasional berfokus pada menjalankan dan memantau sistem untuk memberikan nilai bisnis, dan terus meningkatkan proses dan prosedur. Topik utama termasuk mengelola dan mengotomatisasi perubahan, menanggapi peristiwa, dan mendefinisikan standar untuk berhasil mengelola operasi harian.
- Keamanan: Pilar keamanan berfokus pada perlindungan informasi dan sistem. Topik utama meliputi kerahasiaan dan integritas data, mengidentifikasi dan mengelola siapa yang dapat melakukan apa dengan manajemen izin, melindungi sistem, dan menetapkan kontrol untuk mendeteksi peristiwa keamanan.
- Keandalan: Pilar keandalan berfokus pada kemampuan untuk mencegah, dan dengan cepat pulih dari kegagalan untuk memenuhi permintaan bisnis dan pelanggan. Topik utama mencakup elemen dasar seputar pengaturan, persyaratan lintas proyek, perencanaan pemulihan, dan cara kami menangani perubahan.
- Efisiensi kinerja: Pilar efisiensi kinerja berfokus pada penggunaan TI dan sumber daya komputasi secara efisien. Topik utamanya meliputi pemilihan jenis dan ukuran sumber daya yang tepat berdasarkan persyaratan beban kerja, pemantauan performa, dan pembuatan keputusan berdasarkan informasi untuk mempertahankan efisiensi seiring dengan berkembangnya kebutuhan bisnis.

- Optimalisasi biaya: Pilar optimasi biaya berfokus pada menghindari biaya yang tidak dibutuhkan. Topik utama termasuk memahami dan mengendalikan di mana uang dihabiskan, memilih jumlah jenis sumber daya yang paling tepat dan tepat, menganalisis pengeluaran dari waktu ke waktu, dan penskalaan untuk memenuhi kebutuhan bisnis tanpa pengeluaran berlebihan.
- Keberlanjutan: Pilar keberlanjutan berfokus pada kemampuan untuk terus meningkatkan dampak keberlanjutan dengan mengurangi konsumsi energi dan meningkatkan efisiensi di semua komponen beban kerja dengan memaksimalkan manfaat dari sumber daya yang disediakan dan meminimalkan total sumber daya yang dibutuhkan.

AWS Well-Architected Framework: Menjelaskan AWS bagaimana memungkinkan pelanggan untuk menilai dan meningkatkan arsitektur berbasis cloud mereka dan lebih memahami dampak bisnis dari keputusan desain mereka. Ini membahas prinsip-prinsip desain umum serta praktik terbaik dan panduan khusus dalam enam bidang konseptual yang AWS didefinisikan sebagai pilar Kerangka Well-Architected.

Tanggung jawab lapisan aplikasi vs tanggung jawab lapisan infrastruktur di AMS

Dengan menggunakan AMS, infrastruktur Anda, dan semua yang dibutuhkan untuk pemeliharaan dan pertumbuhan, dikelola oleh AMS. Namun, apa pun yang Anda butuhkan untuk line-of-business aplikasi atau aplikasi produk, dikembangkan, digunakan, dan dikelola oleh Anda.

Dengan bantuan alat penyebaran aplikasi, seperti CodeDeploy dan, atau Chef CloudFormation, Puppet, Ansible, atau Saltstack, penerapan aplikasi Anda ke infrastruktur yang dikelola AMS dapat sepenuhnya otomatis.

Untuk detail tentang apa yang AMS lakukan dan tidak lakukan, lihatApa yang kita lakukan, apa yang tidak kita lakukan.

Mutabilitas EC2 instans Amazon di AMS

Anda dan AMS dapat mempertahankan instans Amazon Elastic Compute Cloud (Amazon EC2) di infrastruktur Anda dengan salah satu dari dua cara:

- Immutable: Model ini menggunakan Amazon Machine Images (AMIs) dipanggang (dibuat) dengan fitur yang diperlukan. Saat menerapkan pembaruan, instance yang ada dirobohkan dan

sepenuhnya diganti dengan yang baru yang dibuat dari AMI yang diperbarui. Untuk meminimalkan waktu henti, proses bergulir ini membuat beberapa contoh tidak diperbarui dan dapat diakses sementara yang lain diperbarui hingga, akhirnya, perubahan baru sepenuhnya diterapkan.

- Mutable: Dalam model ini, infrastruktur diperbarui dengan kode baru yang diterapkan pada sistem yang ada di Cloud. Model ini adalah campuran dari mendorong pembaruan secara manual dan menggunakan infrastructure-as-code untuk menyebarkan pembaruan dan tidak bergantung pada yang baru AMIs.

Model pemeliharaan ini dibahas secara lebih rinci di bagian selanjutnya dari panduan ini.

Menggunakan AWS Secrets Manager dengan sumber daya AMS

Ada banyak kasus di mana Anda mungkin perlu berbagi rahasia dengan AMS, misalnya:

- Reset kata sandi utama untuk contoh RDS
- Sertifikat untuk penyeimbang beban
- Memperoleh kredensi berumur panjang untuk pengguna IAM dari AMS

Cara paling aman untuk berbagi informasi rahasia dengan AMS adalah melalui AWS Secrets Manager; ikuti langkah-langkah berikut:

1. Login ke AWS Konsol menggunakan akses federasi Anda dan CustomerReadOnly peran untuk single-account landing zone (SALZ); gunakan salah satu peran ini,, AWSManaged ServicesSecurityOpsRole AWSManagedServicesAdminRole, dan AWSManaged ServicesChangeManagementRole untuk multi-account landing zone (MALZ).
2. Arahkan ke konsol AWS Secrets manager dan klik Simpan rahasia baru.
3. Pilih “Jenis rahasia lainnya”.
4. Masukkan nilai rahasia sebagai teks biasa dan klik Berikutnya.
5. Masukkan nama dan deskripsi rahasia. Nama harus selalu dimulai dengan "customer-shared/*”. Misalnya "customer-shared/license-2018”. Setelah selesai, lanjutkan dengan mengklik Berikutnya.
6. Gunakan enkripsi KMS default.
7. Biarkan rotasi otomatis dinonaktifkan dan klik Berikutnya.
8. Tinjau dan klik Store, untuk menyimpan rahasia.

9. Balas kami dalam permintaan layanan AMS dengan nama rahasia dan ARN, sehingga kami dapat mengidentifikasi dan mengambil rahasia. Untuk informasi tentang membuat permintaan layanan, lihat Contoh Permintaan Layanan.

Penerapan aplikasi di AMS

Selama orientasi, AWS Managed Services (AMS) bekerja sama dengan Anda untuk menentukan infrastruktur yang Anda butuhkan.

Infrastruktur dasar mencakup AWS virtual private cloud (VPC), keamanan komunikasi melalui trust hutan ADFS, subnet dasar (DMZ, Shared Services, dan Private) yang dicerminkan di dua zona ketersediaan dan dikonfigurasi dengan NAT terkelola, benteng, penyeimbang beban publik, (DX), dan keamanan yang diperlukan. Direct Connect Sumber daya aplikasi Anda akan digunakan di subnet pribadi, atau aplikasi pelanggan Anda. Anda dapat mempelajari lebih lanjut tentang arsitektur AMS tipikal di Panduan Pengguna AWS Managed Services.

Infrastruktur yang Anda terapkan setelah dasar-dasar selesai, harus mencakup semua komponen untuk aplikasi dan pengembangan aplikasi Anda.

Kemampuan penerapan aplikasi di AMS

Beberapa cara Anda dapat menyebarkan aplikasi di AMS. Detail tentang setiap metode mengikuti.

Contoh Kemampuan Penerapan Aplikasi

Nama Metode	Penyebaran Infrastruktur	AMI atau Elemen Kunci	Instal Aplikasi
Aplikasi yang Dapat Diubah, AMS AMI			
Penerapan aplikasi manual	Tumpukan penuh CT atau Tier dan Tie CTs	AMI yang disediakan AMS	Kirim CT manajemen Access, instal aplikasi secara manual.
UserData penyebaran aplikasi dengan agen aplikasi (yaitu Chef, Puppet, dll.)			Gunakan penyediaan CT dengan UserData scripting yang menginstal agen aplikasi, dan yang script/agent menginstal aplikasi.

Nama Metode	Penyebaran Infrastruktur	AMI atau Elemen Kunci	Instal Aplikasi
UserData penyebaran aplikasi tanpa agen (yaitu Ansible, Salt SSH, dll.)			Kirim CT manajemen Akses, instal agen aplikasi. Menyebarkan aplikasi dengan perkakas penerapan aplikasi.
Aplikasi yang Dapat Diubah, AMI Kustom			
Penerapan aplikasi AMI kustom (non-ASG)	Tumpukan penuh CT atau Tier dan Tie CTs	AMI kustom. AMS AMI -> sesuaikan dengan agen perkakas penerapan aplikasi -> buat EC2 instance (CT) -> buat AMI (CT).	Aplikasi menyebarkan perkakas (yaitu Chef), memanfaatkan agen, menyebarkan aplikasi.
Penerapan aplikasi AWS Database Migration Service (DMS)	AWS DMS sinkronisasi ke tumpukan Database Relasional AMS yang ada.	AMI khusus	Pelanggan atau mitra menggunakan AWS Database Migration Service; AMS memverifikasi komponen AMS saat diluncurkan

Nama Metode	Penyebaran Infrastruktur	AMI atau Elemen Kunci	Instal Aplikasi
Penyebaran aplikasi Workload Ingest	Workload Ingest CT yang dimigrasi oleh mitra instance/ AMI dan diprakarsai pelanggan.		Partner memigrasikan instans, membuat AMI di VPC yang dikelola AMS pelanggan; pelanggan menggunakan Workload Ingest CT untuk meluncurkan tumpukan di AMS. Lihat perinciannya di Pencerapan Beban Kerja AMS (WIGS).
Aplikasi yang Tidak Dapat Diubah			
Penerapan aplikasi AMI kustom (ASG)	Tumpukan penuh CT atau Tier dan Tie CTs	AMS AMI -> sesuaikan -> buat EC2 instance (CT) -> buat AMI (CT) -> buat grup Auto Scaling.	Auto Scaling menyebarkan aplikasi dengan AMI kustom Lihat perinciannya di Penyebaran Aplikasi Tier dan Tie di AMS.
Aplikasi yang Dapat Diubah atau Tidak Dapat Diubah			

Nama Metode	Penyebaran Infrastruktur	AMI atau Elemen Kunci	Instal Aplikasi
Penerapan aplikasi CloudFormation Template Kustom	CloudFormation Template	CloudFormation Template AWS -> customize/prepare untuk AMS -> Penerapan | Tertelan | Tumpukan dari CloudFormation Template | Buat (ct-36cn2avfrrj9v).	AMS menerapkan aplikasi Anda ke akun Anda menggunakan CloudFormation template kustom Anda, dan memvalidasi penerapan aplikasi. Lihat perinciannya di AMS CloudFormation menelan.
Impor Database SQL	Operasi AMS (Lainnya | CT lainnya)	Pada basis data SQL premis -> file.bak -> AMS RDS SQL database -> Manajemen | Lainnya | Lainnya | Buat (ct-1e1xtak34nx76) untuk impor.	AMS mengimpor database lokal Anda ke database RDS yang dikelola AMS. Lihat perinciannya di Database (DB) impor ke AMS RDS untuk Microsoft SQL Server.
Database Migration Service (DMS)	Operasi AMS (Beberapa CTs)	Pada basis data premis -> contoh replikasi DMS -> grup subnet replikasi DMS -> Titik akhir target DMS -> Titik akhir sumber DMS -> tugas replikasi DMS.	AMS mengimpor database lokal Anda ke S3 yang dikelola AMS atau database RDS target. Lihat perinciannya di AWS Database Migration Service (AWS DMS).

Nama Metode	Penyebaran Infrastruktur	AMI atau Elemen Kunci	Instal Aplikasi
CodeDeploy penyebaran aplikasi	CodeDeploy	Aplikasi -> CodeDeploy aplikasi -> grup CodeDeploy penyebaran -> CodeDeploy penerapan.	Tergantung pada penggunaan, In-place atau Blue/Green penerapan aplikasi. Lihat perinciannya di CodeDeploy permintaan.

Merencanakan penerapan aplikasi Anda di AMS

Untuk serangkaian pertanyaan yang direkomendasikan untuk dijawab untuk mengaktifkan penerapan aplikasi, lihat. Lampiran: Kuesioner orientasi aplikasi Pertanyaan mencakup menggambarkan Anda:

- Ringkasan penyebaran
- Komponen penyebaran infrastruktur
- Platform hosting aplikasi
- Model penyebaran aplikasi
- Dependensi aplikasi
- Sertifikat SSL untuk aplikasi produk

Pencerapan Beban Kerja AMS (WIGS)

Topik

- Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows
- Bagaimana Migrasi Mengubah Sumber Daya Anda
- Migrasi Beban Kerja: Proses Standar
- Migrasi beban kerja: CloudEndure landing zone (SALZ)
- Akun AMS Tools (memigrasikan beban kerja)
- Memigrasi beban kerja: Validasi pra-konsumsi Linux
- Memigrasi beban kerja: Validasi pra-konsumsi Windows

- Workload Ingest Stack: Membuat

Gunakan tipe perubahan konsumsi beban kerja AMS (CT) dengan mitra migrasi cloud AMS, untuk memindahkan beban kerja yang ada ke VPC yang dikelola AMS. Menggunakan AMS workload ingest, Anda dapat membuat AMS AMI kustom setelah memindahkan instance yang dimigrasi ke AMS. Bagian ini menjelaskan proses, prasyarat, dan langkah-langkah yang diambil oleh mitra migrasi Anda dan diri Anda sendiri untuk menelan beban kerja AMS.

> **Important**
>
> Sistem operasi harus didukung oleh konsumsi beban kerja AMS. Untuk sistem operasi yang didukung, lihatMigrasi Beban Kerja: Prasyarat untuk Linux dan Windows.
> Setiap beban kerja dan akun berbeda. AMS akan bekerja dengan Anda untuk mempersiapkan hasil yang sukses.

Diagram berikut menggambarkan proses konsumsi beban kerja AMS.

Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows

Sebelum memasukkan salinan instans lokal ke AWS Managed Services (AMS), prasyarat tertentu harus dipenuhi. Ini adalah prasyarat, termasuk yang berbeda antara sistem operasi Windows dan Linux.

> **Note**
>
> Untuk menyederhanakan proses penentuan apakah instance siap untuk dikonsumsi, alat validasi untuk Windows dan Linux telah dibuat. Alat ini dapat diunduh dan dijalankan langsung di server lokal Anda serta EC2 instans di AWS. Linux Pra-wig Validation.zip, Windows Pra-wig Validation.zip.

SEBELUM ANDA MULAI, untuk Linux dan Windows:

- Lakukan pemindaian virus penuh.
- Instance harus memiliki profil `customer-mc-ec2-instance-profile` instance.
- Instal Agen Amazon EC2 Systems Manager (SSM) dan pastikan Agen SSM aktif dan berjalan.

- Minimal 10GB ruang disk kosong pada volume root direkomendasikan untuk menjalankan AMS workload ingest (WIGS). Secara operasional, AMS merekomendasikan pemanfaatan disk kurang dari 75% dan memberi peringatan ketika pemanfaatan disk mencapai 85%.
- Tentukan kerangka waktu untuk konsumsi dengan mitra migrasi Anda.
- AMI kustom ada sebagai EC2 instance di akun AMS produksi target (ini adalah tanggung jawab mitra migrasi).

Important

Sistem operasi harus didukung oleh konsumsi beban kerja AMS.

- Sistem operasi berikut ini didukung:
  - Microsoft Windows Server: 2008 R2, 2012, 2012 R2, 2016, 2019 dan 2022
  - Linux: Amazon Linux 2023, Amazon Linux 2, dan Amazon Linux, CentOS 7.x, CentOS 6.5-6.10, Oracle Linux 7: versi minor 7.5 ke atas, Oracle Linux 8: versi minor hingga 8.3, RHEL 8.x, RHEL 7.x, RHEL 6.5-6.10, SUSE Linux Enterprise Server 15,, dan versi khusus SAP, SUSE Linux Enterprise Server 12, Ubuntu 18.04 SP3 SP4 SP5
- AMIs Berikut ini tidak didukung:
  - Amazon Linux 2023 AMI Minimal.

Note

Titik akhir AMS API/CLI (amscm dan amsskms) berada di Wilayah AWS N. Virginia,. `us-east-1` Bergantung pada bagaimana autentikasi Anda disetel, dan di Wilayah AWS akun dan sumber daya Anda, Anda mungkin perlu menambahkan `--region us-east-1` saat mengeluarkan perintah. Anda mungkin juga perlu menambahkan`--profile saml`, jika itu adalah metode otentikasi Anda.

Prasyarat LINUX

Perhatikan persyaratan yang tercantum dalam Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows dan pastikan hal-hal berikut sebelum mengirimkan WIGS RFC:

- Driver jaringan terbaru yang disempurnakan diinstal; lihat Jaringan yang Ditingkatkan di Linux.

- Komponen perangkat lunak pihak ketiga yang akan bertentangan dengan komponen AMS telah dihapus:
  - Klien Anti-Virus
  - Klien Cadangan
  - Perangkat lunak virtualisasi (seperti VM Tools atau layanan Integrasi Hyper-V)
  - Perangkat Lunak Manajemen Akses (Seperti SSSD, Centrify, atau PBIS)
- Pastikan SSH dikonfigurasi dengan benar - Ini untuk sementara memungkinkan otentikasi kunci pribadi untuk SSH. AMS menggunakan ini dengan alat manajemen konfigurasi kami. Gunakan perintah ini:

```
sudo grep -q "^PubkeyAuthentication" /etc/ssh/sshd_config && sudo sed "s/
^PubkeyAuthentication=.*/PubkeyAuthentication yes/" -i /etc/ssh/sshd_config || sudo
 sed "$ a\PubkeyAuthentication yes" -i /etc/ssh/sshd_config
```

```
sudo grep -q "^AuthorizedKeysFile" /etc/ssh/sshd_config && sudo sed "s/
^AuthorizedKeysFile=.*/AuthorizedKeysFile %h\/.ssh\/authorized_keys/" -i /etc/ssh/
sshd_config || sudo sed "$ a\AuthorizedKeysFile %h/.ssh/authorized_keys" -i /etc/ssh/
sshd_config
```

- Pastikan Yum dikonfigurasi dengan benar - RedHat memerlukan lisensi untuk menggunakan Repositori Yum mereka. Instans harus dilisensikan melalui Server Satelit atau RedHat Cloud Server. Gunakan salah satu tautan ini jika lisensi diperlukan:
  - Satelit Red Hat
  - Akses Cloud Red Hat
- Jika Anda menggunakan Red Hat Satellite, WIGS memerlukan penambahan Red Hat Software Collections (RHSCL). Sistem WIGS menggunakan RHSCL untuk menambahkan interpreter Python3.6 bersama apa pun yang dikonfigurasi pada sistem. Untuk mendukung solusi ini, repositori berikut harus tersedia:
  - rhel-server-rhscl
  - rhel-server-releases-optional

Prasyarat Windows

Perhatikan persyaratan yang tercantum dalam Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows dan pastikan hal-hal berikut sebelum mengirimkan WIGS RFC:

- Powershell versi 3 atau lebih tinggi diinstal.
- AWS EC2 Config diinstal pada instance dengan beban kerja yang akan Anda migrasikan.
- Instal driver AWS yang mendukung jenis instans generasi terbaru: PV, ENA, dan NVMe. Anda dapat menggunakan informasi di tautan ini:
  - Memutakhirkan Driver PV pada Instans Windows Anda
  - Jaringan yang Ditingkatkan di Windows
  - NVMe Driver AWS untuk Instans Windows
  - Bagian 3: Memutakhirkan driver AWS NVMe
  - Bagian 5: Memasang Driver Port Serial untuk Instans Bare Metal
  - Bagian 6: Memperbarui Pengaturan Manajemen Daya
- (Opsional tetapi disarankan) Nonaktifkan Layanan kritis — Tetapkan layanan aplikasi penting, seperti database, menjadi dinonaktifkan, tetapi pastikan bahwa setiap perubahan didokumentasikan sehingga dapat dikembalikan ke mode startup aslinya selama tahap verifikasi aplikasi.
- (Opsional tetapi disarankan) Buat AMI Failsafe dari instance yang disiapkan:
  - Menggunakan Deployment | Komponen tumpukan lanjutan | AMI | Buat
  - Selama pembuatan, tambahkan tag Key=Name, Value=Application-id_ IngestReady
  - Tunggu hingga AMI dibuat sebelum melanjutkan
- Komponen perangkat lunak pihak ketiga yang akan bertentangan dengan komponen AMS telah dihapus:
  - Klien Anti-Virus
  - Klien Cadangan
  - Perangkat lunak virtualisasi (seperti VM Tools atau layanan Integrasi Hyper-V)

> **Note**
>
> Program End-of-Support Migrasi untuk server Windows (EMP) mencakup perkakas untuk memigrasikan aplikasi lama Anda dari Windows Server 2003, 2008, dan 2008 R2 ke versi yang lebih baru dan didukung di AWS, tanpa refactoring apa pun.

Bagaimana Migrasi Mengubah Sumber Daya Anda

RFC konsumsi yang dijelaskan di bagian ini mengambil langkah selanjutnya untuk menambahkan konfigurasi ke instance, setelah dimigrasikan ke akun AMS Anda, sehingga AMS dapat mengelolanya.

Konfigurasi yang ditambahkan khusus AMS sebagai berikut.

Perubahan yang dilakukan pada instance Linux yang tertelan:

- Perangkat lunak yang diinstal:
  - Cloud Init: Digunakan untuk mengkonfigurasi kunci pribadi untuk Jarvis Access.
  - Python 3 (bahasa scripting) untuk semua sistem operasi yang didukung (Kecuali untuk CentOS 6, RHEL 8, 7). OracleLinux
  - AWS CloudFormation Python Helper Scripts: CloudFormation AWS menyediakan skrip yang digunakan untuk menginstal perangkat lunak dan memulai layanan pada instans Amazon. EC2
  - AWS CLI: AWS CLI adalah alat open source yang dibangun di atas AWS SDK for Python (Boto) yang menyediakan perintah untuk berinteraksi dengan layanan AWS.
  - AWS SSM Agent: Agen SSM memproses permintaan dari layanan Systems Manager mengonfigurasi mesin seperti yang ditentukan dalam permintaan.
  - AWS CloudWatch Logs Agent: Mengirim log ke CloudWatch.
  - AWS CodeDeploy: Layanan penerapan yang mengotomatiskan penerapan aplikasi ke instans EC2 Amazon, instans lokal, atau fungsi Lambda tanpa server.
  - Ruby: Diperlukan untuk CodeDeploy
  - System Performance Tools (sysstat): Sysstat berisi berbagai utilitas untuk memantau kinerja sistem dan aktivitas penggunaan.
  - AD Bridge (Sebelumnya Layanan PowerBroker Identitas): Bergabung dengan host non-Microsoft ke domain Active Directory.
  - Agen Keamanan Dalam Trend Micro: Perangkat lunak Anti-Virus.
- Perangkat lunak yang diubah:
  - Instans dikonfigurasi untuk menggunakan zona waktu UTC.

Perubahan yang dilakukan pada instance Windows yang dicerna:

- Perangkat lunak yang diinstal:

  - AWS Tools untuk Windows PowerShell: AWS Tools untuk PowerShell memungkinkan pengembang dan administrator mengelola layanan dan sumber daya AWS mereka di lingkungan PowerShell skrip.
  - Agen Keamanan Dalam Trend Micro: Perlindungan Anti-Virus
  - PowerShell Modul AMS yang berisi PowerShell kode untuk mengontrol Boot, Active Directory Join, Monitoring, Security, dan Logging.
- Perangkat lunak yang diubah:
  - Blok Pesan Server (SMB) versi 1 dinonaktifkan.
  - Windows Remote Management (WinRM) diaktifkan dan dikonfigurasi untuk mendengarkan pada port 5986. Aturan firewall yang memungkinkan port masuk ini juga dibuat.
- Perangkat lunak yang mungkin diinstal atau diubah:
  - Microsoft .Net Framework 4.5 (Platform pengembang), jika versi lebih rendah maka .Net Framework 4.5 terdeteksi.
  - Untuk Windows 2012, untuk Windows 2012R2, kami meng-upgrade ke 5.1. PowerShell

Migrasi Beban Kerja: Proses Standar

> **Note**
>
> Karena dua pihak diperlukan untuk proses ini, bagian ini menjelaskan tugas untuk masing-masing: Mitra Migrasi Cloud AMS (mitra migrasi), dan Pemilik Aplikasi (Anda).

1. Mitra migrasi, Siapkan:
   a. Mitra migrasi mengirimkan Permintaan Layanan ke AMS untuk peran IAM untuk tujuan migrasi instans Anda. Untuk detail tentang mengirimkan permintaan layanan, lihat Contoh Permintaan Layanan.
   b. Mitra migrasi mengirimkan Permintaan Akses Admin. Tim Operasi AMS menyediakan akses mitra migrasi ke akun Anda melalui peran IAM yang diminta.
2. Mitra migrasi, Migrasikan Beban Kerja Individu:

a. Mitra migrasi memigrasikan AWS non-instance Anda ke subnet di akun AMS Anda melalui Amazon asli EC2 atau alat migrasi lainnya, dengan profil instans `customer-mc-ec2-instance-profile` IAM (harus ada di akun).

b. Mitra migrasi mengirimkan RFC dengan Deployment | Ingestion | Stack from migration partner migrated instance | Create CT (ct-257p9zjk14ija); untuk detail tentang membuat dan mengirimkan RFC ini, lihat. Workload Ingest Stack: Membuat

   Output eksekusi RFC mengembalikan ID instance, alamat IP, dan ID AMI.

   Mitra migrasi memberi Anda ID instans dari beban kerja yang dibuat di akun Anda.

3. Anda, Mengakses dan Memvalidasi Migrasi:

   a. Dengan menggunakan output eksekusi yang diberikan kepada Anda (ID AMI, ID instans, dan alamat IP) oleh mitra migrasi, kirimkan RFC akses dan masuk ke tumpukan AMS yang baru dibuat dan verifikasi bahwa aplikasi Anda berfungsi dengan baik. Untuk detailnya, lihat Meminta Akses Instance.

   b. Jika puas, Anda dapat terus menggunakan instance yang diluncurkan sebagai tumpukan 1 tingkat and/or menggunakan AMI untuk membuat tumpukan tambahan, termasuk grup Auto Scaling.

   c. Jika tidak puas dengan migrasi, ajukan permintaan layanan dan referensikan tumpukan dan RFC IDs; AMS akan bekerja sama dengan Anda untuk mengatasi masalah Anda.

CloudEndure proses pengambilan beban kerja landing zone dijelaskan selanjutnya.

Migrasi beban kerja: CloudEndure landing zone (SALZ)

Bagian ini memberikan informasi tentang pengaturan migrasi perantara single-account landing zone (SALZ) untuk instance cutover CloudEndure (CE) agar tersedia untuk RFC workload ingest (WIGS).

Untuk mempelajari selengkapnya CloudEndure, lihat CloudEndure Migrasi.

> **Note**
>
> Ini adalah standar, keamanan diperkeras, migrasi LZ dan pola.

Prasyarat:

- Akun AMS pelanggan
- Integrasi jaringan dan akses antara akun AMS dan pelanggan di lokasi
- Sebuah CloudEndure akun
- Alur kerja pra-persetujuan untuk peninjauan dan penandatanganan AMS Security, dijalankan dengan CA and/or CSDM Anda, (misalnya, penyalahgunaan kredenal permanen pengguna IAM memberikan kemampuan untuk instans dan grup keamanan) create/delete

> **Note**
>
> Persiapan khusus dan proses migrasi dijelaskan di bagian ini.

Persiapan: Anda dan operator AMS:

1. Siapkan Permintaan Perubahan (RFC) dengan Manajemen | Lainnya | Lainnya | Perbarui jenis perubahan ke AMS untuk sumber daya dan pembaruan berikut. Anda dapat mengirimkan terpisah Lainnya | Pembaruan Lainnya RFCs, atau satu. Untuk detail tentang RFC/CT itu, lihat Lainnya | Pembaruan Lainnya dengan permintaan ini:

    a. Tetapkan blok CIDR sekunder di VPC AMS Anda; blok CIDR sementara yang akan dihapus setelah migrasi selesai. Pastikan bahwa blok tidak akan bertentangan dengan rute yang ada kembali ke jaringan lokal Anda. Misalnya, jika CIDR VPC AMS Anda adalah 10.0.0.0/16, dan ada rute kembali ke netword on-premise Anda 10.1.0.0/16, maka CIDR sekunder sementara bisa 10.255.255.0/24. Untuk informasi tentang blok AWS CIDR, lihat Ukuran VPC dan Subnet.

    b. Buat subnet baru, pribadi, di dalam VPC AMS taman awal. Contoh nama:`migration-temp-subnet`.

    c. Buat tabel rute baru untuk subnet dengan hanya rute VPC dan NAT (Internet) lokal, untuk menghindari konflik dengan server sumber selama pemotongan instance dan kemungkinan pemadaman. Pastikan lalu lintas keluar ke Internet diizinkan untuk unduhan patch, sehingga prasyarat AMS WIGS dapat diunduh dan diinstal.

    d. Perbarui grup keamanan iklan terkelola untuk mengizinkan lalu lintas masuk dan keluar. to/from `migration-temp-subnet` Juga minta agar grup keamanan EPS load balancer (ELB)

Anda (mis:`mc-eps-McEpsElbPrivateSecurityGroup-M790XBZEEX74`) diperbarui untuk mengizinkan subnet baru, pribadi, (mis.). `migration-temp-subnet` Jika lalu lintas dari subnet khusus CloudEndure (CE) tidak diizinkan pada ketiga port TCP, konsumsi WIGS akan gagal.

e. Terakhir, minta kebijakan CloudEndure IAM baru dan pengguna IAM. <Customer Application Subnet (s) + Temp Migration Subnet>Kebijakan ini membutuhkan nomor akun Anda yang benar, dan subnet IDs dalam `RunInstances` pernyataan harus: milik Anda.

   Untuk melihat CloudEndure kebijakan IAM AMS yang telah disetujui sebelumnya: Buka paket file Contoh Zona Pendaratan WIGS Cloud Endure dan buka file. `customer_cloud_endure_policy.json`

> **Note**
>
> Jika Anda menginginkan kebijakan yang lebih permisif, diskusikan apa yang Anda butuhkan dengan Anda CloudArchitect/CSDM dan dapatkan, jika diperlukan, AMS Security Review and Signoff sebelum mengirimkan RFC yang menerapkan kebijakan tersebut.

2. Langkah-langkah persiapan Anda yang akan digunakan CloudEndure untuk menelan beban kerja AMS telah selesai dan, jika mitra migrasi Anda telah menyelesaikan langkah persiapannya, migrasi siap dilakukan. WIGS RFC dikirimkan oleh mitra migrasi Anda.

> **Note**
>
> Kunci pengguna IAM tidak akan langsung dibagikan, tetapi harus diketik ke konsol CloudEndure manajemen oleh operator AMS dalam sesi berbagi layar.

Persiapan: Mitra Migrasi dan Operator AMS:

1. Buat proyek CloudEndure migrasi.

   a. Selama pembuatan proyek, miliki kredensi pengguna IAM yang mengetik AMS dalam sesi berbagi layar.

   b. Di Pengaturan Replikasi -> Pilih subnet tempat Server Replikasi akan diluncurkan, pilih subnet. customer-application-x

c. Di Pengaturan Replikasi -> Pilih Grup Keamanan untuk diterapkan ke Server Replikasi, pilih kedua grup keamanan Sentinel (Hanya Pribadi dan). EgressAll

2. Tentukan opsi cutover untuk mesin (instance).

   a. Subnet: migration-temp-subnet.

   b. Grup Keamanan: Kedua grup keamanan “Sentinel” (Hanya Pribadi dan EgressAll).

      Instans cutover harus dapat berkomunikasi dengan AMS Managed AD dan ke titik akhir publik AWS.

   c. IP elastis: Tidak ada

   d. IP Publik: tidak

   e. Peran IAM: customer-mc-ec 2-instance-profile

      Peran IAM harus memungkinkan komunikasi SSM. Lebih baik menggunakan AMS default.

   f. Tetapkan tag sesuai konvensi.

Migrasi: Mitra Migrasi:

1. Buat tumpukan dummy di AMS. Anda menggunakan ID tumpukan untuk mendapatkan akses ke benteng.
2. Instal agen CloudEndure (CE) di server sumber. Untuk detailnya, lihat Menginstal Agen.
3. Buat kredensi admin lokal di server sumber.
4. Jadwalkan jendela cutover pendek dan klik Cutover, saat siap. Ini menyelesaikan migrasi dan mengarahkan pengguna ke Wilayah AWS target.
5. Minta tumpukan akses Admin ke tumpukan dummy, lihat Permintaan Akses Admin.
6. Masuk ke benteng, lalu ke instance cutover menggunakan kredenal admin lokal yang Anda buat.
7. Buat AMI failsafe. Untuk detail tentang pembuatan AMIs, lihat AMI Create.
8. Siapkan contoh untuk konsumsi, lihat. Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows
9. Jalankan WIGS RFC terhadap instance, lihat. Workload Ingest Stack: Membuat

Akun AMS Tools (memigrasikan beban kerja)

Akun alat Zona Pendaratan Multi-Akun Anda (dengan VPC) membantu mempercepat upaya migrasi, meningkatkan posisi keamanan Anda, mengurangi biaya dan kompleksitas, dan menstandarisasi pola penggunaan Anda.

Akun alat menyediakan yang berikut:

- Batas yang terdefinisi dengan baik untuk akses ke instance replikasi untuk integrator sistem di luar beban kerja produksi Anda.
- Memungkinkan Anda membuat ruang terisolasi untuk memeriksa beban kerja untuk malware, atau rute jaringan yang tidak dikenal, sebelum menempatkannya ke akun dengan beban kerja lainnya.
- Sebagai pengaturan akun yang ditentukan, ini menyediakan waktu yang lebih cepat untuk onboard dan menyiapkan untuk memigrasi beban kerja.
- Rute jaringan terisolasi untuk mengamankan lalu lintas dari on-premise -> -> Tools account CloudEndure -> AMS menelan gambar. Setelah gambar tertelan, Anda dapat membagikan gambar ke akun tujuan melalui Manajemen AMS | Komponen tumpukan lanjutan | AMI | Bagikan (ct-1eiczxw8ihc18) RFC.

Diagram arsitektur tingkat tinggi:

Gunakan Deployment | Managed landing zone | Akun manajemen | Buat akun alat (dengan VPC) jenis perubahan (ct-2j7q1hgf26x5c), untuk dengan cepat menyebarkan akun alat dan membuat instance proses Penyerapan Beban Kerja dalam lingkungan Zona Pendaratan Multi-Akun. Lihat Akun manajemen, Akun alat: Membuat (dengan VPC).

> **Note**
>
> Kami merekomendasikan memiliki dua zona ketersediaan (AZs), karena ini adalah hub migrasi.
> Secara default, AMS membuat dua grup keamanan berikut (SGs) di setiap akun.
> Konfirmasikan bahwa SGs keduanya hadir. Jika mereka tidak hadir, silakan buka permintaan layanan baru dengan tim AMS untuk memintanya.
>
> - SentinelDefaultSecurityGroupPrivateOnlyEgressAll
> - InitialGarden-SentinelDefaultSecurityGroupPrivateOnly

Pastikan bahwa instance CloudEndure replikasi dibuat di subnet pribadi di mana ada rute kembali ke on-premise. Anda dapat mengonfirmasi bahwa dengan memastikan bahwa tabel rute untuk subnet pribadi memiliki rute default kembali ke TGW. Namun, melakukan pemotongan CloudEndure mesin harus masuk ke subnet pribadi “terisolasi” di mana tidak ada rute kembali ke on-premise, hanya lalu lintas keluar Internet yang diizinkan. Sangat penting untuk memastikan cutover terjadi di subnet yang terisolasi untuk menghindari potensi masalah pada sumber daya di lokasi.

Prasyarat:

1. Baik tingkat dukungan Plus atau Premium.
2. Akun aplikasi IDs untuk kunci KMS tempat AMIs dikerahkan.
3. Akun alat, dibuat seperti yang dijelaskan sebelumnya.

AWS Layanan Migrasi Aplikasi (AWS MGN)

AWS Layanan Migrasi Aplikasi (AWS MGN) dapat digunakan di akun MALZ Tools Anda melalui peran `AWSManagedServicesMigrationRole` IAM yang dibuat secara otomatis selama penyediaan akun Tools. Anda dapat menggunakan AWS MGN untuk memigrasikan aplikasi dan database yang berjalan pada versi sistem operasi Windows dan Linux yang didukung.

Untuk up-to-date informasi paling banyak tentang AWS Region dukungan, lihat Daftar Layanan AWS Regional.

Jika pilihan Anda saat AWS Region ini tidak didukung oleh AWS MGN, atau sistem operasi tempat aplikasi Anda berjalan saat ini tidak didukung oleh AWS MGN, pertimbangkan untuk menggunakan akun CloudEndure Migrasi di Alat Anda.

Meminta Inisialisasi AWS MGN

AWS MGN harus diinisialisasi oleh AMS sebelum digunakan pertama kali. Untuk meminta ini untuk akun Tools baru, kirimkan Manajemen | Lainnya | RFC lain dari akun Tools dengan rincian berikut:

```
RFC Subject=Please initialize AWS MGN in this account
RFC Comment=Please click 'Get started' on the MGN welcome page here:
```

```
    https://console.aws.amazon.com/mgn/home?region=MALZ_PRIMARY_REGION#/welcome using
 all default values
    to 'Create template' and complete the initialization process.
```

Setelah AMS berhasil menyelesaikan RFC dan menginisialisasi AWS MGN di akun Tools Anda, Anda dapat menggunakannya `AWSManagedServicesMigrationRole` untuk mengedit template default untuk kebutuhan Anda.

Aktifkan akses ke akun AMS Tools baru

Setelah akun alat dibuat, AMS memberi Anda ID akun. Langkah Anda selanjutnya adalah mengonfigurasi akses ke akun baru. Ikuti langkah-langkah ini.

1. Perbarui grup Direktori Aktif yang sesuai ke akun yang sesuai IDs.

   Akun baru yang dibuat AMS disediakan dengan kebijakan ReadOnly peran serta peran untuk memungkinkan pengguna mengajukan. RFCs

   Akun Tools juga memiliki peran IAM tambahan dan pengguna yang tersedia:

   - IAM role: `AWSManagedServicesMigrationRole`
   - Pengguna IAM: `customer_cloud_endure_user`
2. Minta kebijakan dan peran untuk memungkinkan anggota tim integrasi layanan menyiapkan alat tingkat berikutnya.

   Arahkan ke konsol AMS dan ajukan yang berikut ini RFCs:

   a. Buat kunci KMS. Gunakan Create KMS Key (auto) atau Create KMS Key (review required).

      Saat Anda menggunakan KMS untuk mengenkripsi sumber daya yang tertelan, menggunakan satu kunci KMS yang dibagikan dengan akun aplikasi Zona Pendaratan Multi-Akun lainnya, memberikan keamanan untuk gambar yang dicerna di mana mereka dapat didekripsi di akun tujuan.

   b. Bagikan kunci KMS.

      Gunakan Manajemen | Komponen tumpukan lanjutan | Kunci KMS | Bagikan (perlu ditinjau) jenis perubahan (ct-05yb337abq3x5) untuk meminta agar kunci KMS baru dibagikan dengan akun aplikasi Anda di mana tertelan akan berada. AMIs

Contoh grafik pengaturan akun akhir:

Contoh kebijakan IAM yang telah disetujui AMS CloudEndure

Untuk melihat CloudEndure kebijakan IAM AMS yang telah disetujui sebelumnya:
Buka paket file Contoh Zona Pendaratan Cloud Endure WIGS dan buka file.
`customer_cloud_endure_policy.json`

Menguji konektivitas dan end-to-end penyiapan akun AMS Tools

1. Mulailah dengan mengkonfigurasi CloudEndure dan menginstal CloudEndure agen di server yang akan mereplikasi ke AMS.
2. Buat proyek di CloudEndure.
3. Masukkan AWS kredensi yang dibagikan saat Anda melakukan prasyarat, meskipun manajer rahasia.
4. Dalam pengaturan Replikasi:
   a. Pilih kedua grup keamanan AMS “Sentinel” (Hanya Pribadi dan EgressAll) untuk Pilih Grup Keamanan untuk diterapkan ke opsi Server Replikasi.
   b. Tentukan opsi cutover untuk mesin (instance). Untuk informasi lebih lanjut, lihat Langkah 5. Potong
   c. Subnet: Subnet pribadi.
5. Grup Keamanan:
   a. Pilih kedua grup keamanan AMS “Sentinel” (Hanya Pribadi dan EgressAll).
   b. Instans cutover harus berkomunikasi ke Active Directory (MAD) yang dikelola AMS dan ke titik akhir publik: AWS
      i. IP elastis: Tidak ada
      ii. IP Publik: tidak
      iii. Peran IAM: customer-mc-ec 2-instance-profile
   c. Tetapkan tag sesuai konvensi penandaan internal Anda.
6. Instal CloudEndure agen di mesin dan cari contoh replikasi yang muncul di akun AMS Anda di EC2 konsol.

Proses konsumsi AMS:

Kebersihan akun AMS Tools

Anda akan ingin membersihkan setelah Anda selesai di akun telah membagikan AMI dan tidak lagi membutuhkan instance yang direplikasi:

- Posting WIGs konsumsi contoh:
  - Instans cutover: Minimal, hentikan atau hentikan instance ini, setelah pekerjaan selesai, melalui konsol AWS
  - Pencadangan AMI Pra-Penyerapan: Hapus setelah instance tertelan dan instance di lokasi dihentikan
  - Instans yang dicerna AMS: Matikan tumpukan atau hentikan setelah AMI dibagikan
  - AMS-ingested AMIs: Hapus setelah berbagi dengan akun tujuan selesai
- Pembersihan akhir migrasi: Dokumentasikan sumber daya yang digunakan melalui mode Pengembang untuk memastikan pembersihan terjadi secara teratur, misalnya:
  - Grup keamanan
  - Sumber daya yang dibuat melalui Formasi Cloud
  - Jaringan ACK
  - Subnet
  - VPC
  - Tabel Rute
  - Peran
  - Pengguna dan akun

Migrasi dalam skala - Pabrik Migrasi

Lihat Memperkenalkan Solusi Pabrik CloudEndure Migrasi AWS.

Memigrasi beban kerja: Validasi pra-konsumsi Linux

Anda dapat memvalidasi bahwa instans Anda siap untuk dimasukkan ke dalam akun AMS Anda. Validasi pra-konsumsi beban kerja (WIGS) melakukan pemeriksaan seperti jenis sistem operasi, ruang disk yang tersedia, keberadaan perangkat lunak pihak ketiga yang bertentangan, dll. Saat dijalankan, validasi pra-konsumsi WIGS menghasilkan tabel di layar, dengan file log opsional.

Hasilnya memberikan pass/fail status untuk setiap pemeriksaan validasi bersama dengan alasan kegagalan. Selain itu, Anda dapat menyesuaikan tes validasi sesuai dengan kebutuhan Anda.

Pertanyaan yang sering diajukan:

- Bagaimana cara menggunakan validasi pra-konsumsi WIGS Linux?

  Ikuti langkah-langkah berikut untuk mengunduh dan menggunakan skrip validasi pra-konsumsi AMS Linux WIGS:

  1. Unduh file ZIP dengan skrip validasi

     Linux WIGS File zip Validasi Pra-konsumsi.

  2. Buka zip aturan terlampir ke direktori pilihan Anda.
  3. Ikuti instruksi dalam file readme.md.
- Validasi apa yang dilakukan oleh validasi pra-konsumsi WIGS Linux?

  Solusi validasi pra-konsumsi AMS Linux WIGS memvalidasi hal-hal berikut:

  1. Setidaknya ada 5 Gigabyte gratis pada volume boot.
  2. Sistem operasi didukung oleh AMS.
  3. Instance memiliki profil instance tertentu.
  4. Instans tidak mengandung perangkat lunak Antivirus atau perangkat lunak Virtualisasi.
  5. SSH dikonfigurasi dengan benar.
  6. Instance memiliki akses ke Yum Repositories.
  7. Driver jaringan yang ditingkatkan diinstal.
  8. Instans memiliki Agen SSM dan sedang berjalan.
- Mengapa ada dukungan untuk file konfigurasi khusus?

  Skrip dirancang untuk berjalan di server fisik di lokasi dan pada instans AWS. EC2 Namun, seperti yang ditunjukkan dalam daftar di atas, beberapa pengujian akan gagal saat dijalankan di tempat. Misalnya, server fisik di pusat data tidak akan memiliki profil instance. Dalam kasus seperti ini, Anda dapat mengedit file konfigurasi untuk melewati tes profil instance untuk menghindari kebingungan.
- Bagaimana cara memastikan saya memiliki versi terbaru dari skrip?

  up-to-dateVersi solusi Linux WIGS Pre-Ingestion Validation akan tersedia di bagian AMS Helper Files pada halaman Dokumentasi utama.

- Apakah skripnya hanya-baca?

  Skrip dirancang agar hanya-baca kecuali untuk file log yang dihasilkannya, tetapi praktik terbaik harus diikuti untuk menjalankan skrip di lingkungan non-produksi.

- Apakah validasi pra-konsumsi WIGS tersedia untuk Windows?

  Ya. Ini tersedia di bawah bagian AMS Helper Files di halaman Dokumentasi utama.

Memigrasi beban kerja: Validasi pra-konsumsi Windows

Anda dapat menggunakan skrip pra WIGs validator untuk memvalidasi bahwa instans Anda siap untuk dimasukkan ke dalam akun AMS Anda. Validasi pra-konsumsi beban kerja (WIGS) melakukan pemeriksaan seperti jenis sistem operasi, ruang disk yang tersedia, keberadaan perangkat lunak pihak ketiga yang saling bertentangan, dan sebagainya. Saat dijalankan, validasi pra-konsumsi WIGS menghasilkan tabel di layar dan file log opsional. Hasilnya memberikan pass/fail status untuk setiap pemeriksaan validasi bersama dengan alasan kegagalan. Selain itu, Anda dapat menyesuaikan tes validasi.

Pertanyaan yang sering diajukan:

- Bagaimana cara menggunakan validasi pra-konsumsi Windows WIGS?

  Anda dapat menjalankan validasi dari GUI dan browser web, atau Anda dapat menggunakan Windows, SSM Run Command PowerShell, atau SSM Session Manager.

  Opsi 1: Jalankan dari GUI dan browser web

  Untuk menjalankan pra- WIGs validasi Windows dari GUI dan browser web, lakukan hal berikut:

  1. Unduh file ZIP dengan skrip validasi:

     Windows WIGS File ZIP Validasi Pra-konsumsi.

  2. Buka zip aturan terlampir ke direktori pilihan Anda.
  3. Ikuti instruksi dalam file README.md.

  Opsi 2: Jalankan dari Windows PowerShell, SSM Run Command, atau SSM Session Manager

  Windows 2016 dan yang lebih baru

  1. Unduh file ZIP dengan skrip validasi.

```
$DestinationFile = "$env:TEMP\WIGValidation.zip"

$Bucket = 'https://docs.aws.amazon.com/managedservices/latest/appguide/samples/
windows-prewigs-validation.zip'
$DestinationFile = "$env:TEMP\WIGValidation.zip"
$ScriptFolder = "$env:TEMP\AWSManagedServices.PreWigs.Validation"
```

2. Hapus file yang ada dari`C:\Users\AppData\Local\Temp\AWSManagedServices.PreWigs.Validation`.

```
Remove-Item $scriptFolder -Recurse -Force -ErrorAction Ignore
```

3. Memanggil skrip.

```
Invoke-WebRequest -Uri $bucket -OutFile $DestinationFile
Add-Type -Assembly "system.io.compression.filesystem"
```

4. Buka zip file terlampir ke direktori pilihan Anda.

```
[io.compression.zipfile]::ExtractToDirectory($DestinationFile, $env:TEMP)
```

5. Jalankan skrip validasi secara interaktif dan lihat hasilnya.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation -RunWithoutExitCodes
```

6. (Opsional) Untuk menangkap kode kesalahan yang tercantum di bagian Kode Keluar, jalankan skrip tanpa `RunWithoutExitCodes` opsi. Perhatikan bahwa perintah ini mengakhiri PowerShell sesi aktif.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation
```

Windows 2012 R2 dan sebelumnya

Jika Anda menjalankan Windows Server 2012R2 atau di bawahnya, Anda harus mengatur TLS sebelum mengunduh file zip. Untuk mengatur TLS, selesaikan langkah-langkah berikut:

1. Unduh file ZIP dengan skrip validasi.

```
$DestinationFile = "$env:TEMP\WIGValidation.zip"
```

```
$Bucket = 'https://docs.aws.amazon.com/managedservices/latest/appguide/samples/
windows-prewigs-validation.zip'
$DestinationFile = "$env:TEMP\WIGValidation.zip"
$ScriptFolder = "$env:TEMP\AWSManagedServices.PreWigs.Validation"
```

2. Jika ada file validasi yang ada, maka hapus.

```
Remove-Item $scriptFolder -Recurse -Force -ErrorAction Ignore
```

3. Atur versi TLS.

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
```

4. Unduh validasi WIG.

```
Invoke-WebRequest -Uri $bucket -OutFile $DestinationFile
Add-Type -Assembly "system.io.compression.filesystem"
```

5. Buka zip aturan terlampir ke direktori pilihan Anda.

```
[io.compression.zipfile]::ExtractToDirectory($DestinationFile, $env:TEMP)
```

6. Jalankan skrip validasi secara interaktif dan lihat hasilnya.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation -RunWithoutExitCodes
```

7. (Opsional) Untuk menangkap kode kesalahan yang tercantum di bagian Kode Keluar, jalankan skrip tanpa RunWithoutExitCodes opsi. Perhatikan bahwa perintah ini mengakhiri PowerShell sesi aktif.

```
Import-Module .\AWSManagedServices.PreWigs.Validation.psm1 -force
Invoke-PreWIGsValidation
```

> **Note**
>
> Anda dapat mengunduh dan menjalankan PowerShell skrip. Untuk melakukan ini, unduh pre-wigs-validation-powershell-scripts.zip.

- Validasi apa yang dilakukan oleh Windows WIGS Pra-Ingestion Validasi?

Solusi validasi pra-konsumsi AMS Windows WIGS memvalidasi hal-hal berikut:

1. Setidaknya ada 10 Gigabytes gratis pada volume boot.
2. Sistem operasi didukung oleh AMS.
3. Instance memiliki profil instance tertentu.
4. Instans tidak mengandung perangkat lunak antivirus atau perangkat lunak virtualisasi.
5. DHCP diaktifkan pada setidaknya satu adaptor jaringan.
6. Instance siap untuk Sysprep.
   - Untuk 2008 R2 dan 2012 Base dan R2, Sysprep memverifikasi bahwa:
     - Ada file unattend.xml
     - File sppnp.dll (jika ada) tidak rusak
     - Sistem Operasi Belum Diupgrade
     - Sysprep belum berjalan lebih dari jumlah maksimum kali per pedoman Microsoft
   - Untuk 2016 dan di atas, semua pemeriksaan di atas dilewati karena tidak menyebabkan masalah untuk OS itu
7. Subsistem instrumentasi manajemen Windows (WMI) sehat.
8. Driver yang diperlukan diinstal.
9. Agen SSM dan diinstal dan dijalankan.
10. Peringatan diberikan untuk memverifikasi apakah mesin dalam masa tenggang karena Konfigurasi Lisensi RDS.
11. Kunci registri yang diperlukan diatur dengan benar. Untuk detail selengkapnya, lihat README di file zip Validasi Pra-konsumsi.

- Mengapa ada dukungan untuk file konfigurasi khusus?

  Skrip dirancang untuk berjalan di server fisik di lokasi dan pada instans AWS. EC2 Namun, seperti yang ditunjukkan dalam daftar di atas, beberapa pengujian akan gagal saat dijalankan di tempat. Misalnya, server fisik di pusat data tidak akan memiliki profil instance. Dalam kasus seperti ini, Anda dapat mengedit file konfigurasi untuk melewati tes profil instance untuk menghindari kebingungan.

- Bagaimana cara memastikan saya memiliki versi skrip terbaru?

  up-to-dateVersi solusi validasi pra-konsumsi Windows WIGS akan tersedia di bawah bagian File Pembantu AMS di halaman Dokumentasi utama.

- Apakah skripnya hanya-baca?

  Skrip dirancang untuk hanya baca kecuali untuk file log yang dihasilkannya, tetapi praktik terbaik harus diikuti untuk menjalankan skrip di lingkungan non-produksi.

- Apakah Validasi Pra-Ingestion WIGS tersedia untuk Linux?

  Ya. Versi Linux diluncurkan pada 31 Oktober 2019. Ini tersedia di bawah bagian AMS Helper Files di halaman Dokumentasi utama.

Workload Ingest Stack: Membuat

Memigrasi instance ke tumpukan AMS dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.

4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

> **Note**
>
> Jika RFC ditolak, output eksekusi menyertakan tautan ke CloudWatch log Amazon. AMS Workload Ingest (WIGS) RFCs ditolak ketika persyaratan tidak terpenuhi; misalnya, jika perangkat lunak anti-virus terdeteksi pada instance. CloudWatch Log akan mencakup informasi tentang persyaratan yang gagal dan tindakan yang harus diambil untuk perbaikan.

Memigrasi instance ke tumpukan AMS dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

Anda dapat menggunakan AMS CLI untuk membuat instance AMS dari instans non-AMS yang dimigrasikan ke akun AMS.

> **Note**
>
> Pastikan Anda telah mengikuti prasyarat; lihat Memigrasi Beban Kerja: Prasyarat untuk Linux dan Windows.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc --change-type-id "ct-257p9zjk14ija" --change-type-version "2.0" --
title "AMS-WIG-TEST-NO-ACTION" --execution-parameters "{\"InstanceId\":\"INSTANCE_ID\",
\"TargetVpcId\":\"VPC_ID\",\"TargetSubnetId\":\"SUBNET_ID\",\"TargetInstanceType\":
\"t2.large\",\"ApplyInstanceValidation\":true,\"Name\":\"WIG-TEST\",\"Description\":
\"WIG-TEST\",\"EnforceIMDSV2\":\"false\"}"
```

TEMPLATE MEMBUAT:

1. 0utput parameter eksekusi skema JSON untuk jenis perubahan ini ke file; contoh menamainya .json: MigrateStackParams

```
aws amscm get-change-type-version --change-type-id "ct-257p9zjk14ija" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > MigrateStackParams.json
```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"InstanceId":          "MIGRATED_INSTANCE_ID",
"TargetVpcId":         "VPC_ID",
"TargetSubnetId":      "SUBNET_ID",
"Name":                "Migrated-Stack",
"Description":         "Create-Migrated-Stack",
"EnforceIMDSV2":       "false"
}
```

3. Keluarkan file JSON template RFC; contoh menamainya MigrateStackRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > MigrateStackRfc.json
```

4. Ubah dan simpan MigrateStackRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeId":         "ct-257p9zjk14ija",
"ChangeTypeVersion":    "2.0",
"Title":                "Migrate-Stack-RFC"
}
```

5. Buat RFC, tentukan MigrateStackRfc file dan file: MigrateStackParams

```
aws amscm create-rfc --cli-input-json file://MigrateStackRfc.json  --execution-
parameters file://MigrateStackParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Instance baru muncul di daftar Instans untuk akun pemilik aplikasi untuk VPC yang relevan.

6. Setelah RFC berhasil diselesaikan, beri tahu pemilik aplikasi sehingga ia dapat masuk ke instance baru dan memverifikasi bahwa beban kerja operasional.

> **Note**
>
> Jika RFC ditolak, output eksekusi menyertakan tautan ke CloudWatch log Amazon. AMS Workload Ingest (WIGS) RFCs ditolak ketika persyaratan tidak terpenuhi; misalnya, jika perangkat lunak anti-virus terdeteksi pada instance. CloudWatch Log akan mencakup informasi tentang persyaratan yang gagal dan tindakan yang harus diambil untuk perbaikan.

Kiat

> **Note**
>
> Pastikan Anda telah mengikuti prasyarat; lihat Memigrasi Beban Kerja: Prasyarat untuk Linux dan Windows.

> **Note**
>
> Jika tag pada instance yang dimigrasikan memiliki kunci yang sama dengan tag yang disediakan di RFC, RFC akan gagal.

> **Note**
>
> Anda dapat menentukan hingga empat Target IDs, Port, dan Availability Zones.

> **Note**
>
> Jika RFC ditolak, output eksekusi menyertakan tautan ke CloudWatch log Amazon. AMS Workload Ingest (WIGS) RFCs ditolak ketika persyaratan tidak terpenuhi; misalnya, jika perangkat lunak anti-virus terdeteksi pada instance. CloudWatch Log akan mencakup informasi tentang persyaratan yang gagal dan tindakan yang harus diambil untuk perbaikan.

> **Note**
>
> Jika RFC ditolak, output eksekusi menyertakan tautan ke CloudWatch log Amazon. AMS Workload Ingest (WIGS) RFCs ditolak ketika persyaratan tidak terpenuhi; misalnya, jika perangkat lunak anti-virus terdeteksi pada instance. CloudWatch Log akan mencakup informasi tentang persyaratan yang gagal dan tindakan yang harus diambil untuk perbaikan.

Jika perlu, lihat Kegagalan konsumsi beban kerja (WIGS).

AMS CloudFormation menelan

AMS AWS CloudFormation ingest change type (CT) memungkinkan Anda menggunakan CloudFormation templat yang ada, dengan beberapa modifikasi, untuk menerapkan tumpukan khusus dalam VPC yang dikelola AMS.

Topik

- CloudFormation Pedoman Ingest, Praktik Terbaik, dan Batasan
- CloudFormation Ingest: Contoh
- Buat CloudFormation tumpukan ingest
- Perbarui CloudFormation tumpukan ingest
- Menyetujui set perubahan CloudFormation tumpukan ingest
- Perbarui CloudFormation perlindungan penghentian tumpukan
- Penerapan IAM otomatis menggunakan ingest CFN atau pembaruan tumpukan di AMS CTs

Proses CloudFormation menelan AMS melibatkan hal-hal berikut:

- Siapkan dan unggah CloudFormation template kustom Anda ke bucket S3, atau berikan templat sebaris saat membuat RFC. Jika Anda menggunakan bucket S3 dengan URL yang telah ditetapkan sebelumnya; untuk informasi selengkapnya, lihat presign.
- Kirimkan jenis perubahan CloudFormation ingest ke AMS di RFC. Untuk panduan jenis perubahan ingest CFN, lihat. Buat CloudFormation tumpukan ingest Untuk contoh konsumsi CFN, lihat. CloudFormation Ingest: Contoh

- Setelah tumpukan Anda dibuat, Anda dapat memperbaruinya, dan memulihkan penyimpangan di atasnya; Selain itu, jika pembaruan gagal, Anda dapat secara eksplisit menyetujui dan mengimplementasikan pembaruan. Semua prosedur ini dijelaskan dalam bagian ini.

  Untuk informasi tentang deteksi drift CFN, lihat New — CloudFormation Drift Detection.

> **Note**
>
> - Jenis perubahan ini sekarang memiliki versi 2.0. Versi 2.0 otomatis; tidak dieksekusi secara manual. Ini memungkinkan eksekusi CT berjalan lebih cepat. Dua parameter baru diperkenalkan dengan versi ini: CloudFormationTemplate, yang memungkinkan Anda menempelkan CloudFormation template khusus ke RFC, dan VpcId, yang memungkinkan CloudFormation ingest digunakan dengan landing zone multi-akun AMS.
> - Versi 1.0 adalah jenis perubahan manual. Ini berarti bahwa operator AMS harus mengambil beberapa tindakan sebelum jenis perubahan berhasil disimpulkan. Minimal, ulasan diperlukan. Versi ini juga memerlukan nilai parameter CloudFormationTemplateS3Endpoint menjadi URL yang telah ditandatangani sebelumnya.

CloudFormation Pedoman Ingest, Praktik Terbaik, dan Batasan

Agar AMS dapat memproses CloudFormation template Anda, ada beberapa pedoman dan batasan.

Pedoman

Untuk mengurangi CloudFormation kesalahan saat melakukan CloudFormation ingest, ikuti panduan ini:

- Jangan menyematkan kredensi atau informasi sensitif lainnya di template — Template terlihat di CloudFormation konsol, jadi Anda tidak ingin menyematkan kredensi atau data sensitif di CloudFormation templat. Template tidak dapat berisi informasi sensitif. Sumber daya berikut hanya diperbolehkan jika Anda menggunakan AWS Secrets Manager untuk nilainya:
  - `AWS::RDS::DBInstance` - [MasterUserPassword,TdeCredentialPassword]
  - `AWS::RDS::DBCluster` - [MasterUserPassword]
  - `AWS::ElastiCache::ReplicationGroup` - [AuthToken]

> **Note**
>
> Untuk informasi tentang menggunakan AWS rahasia Secrets Manager di properti resource, lihat Cara membuat dan mengambil rahasia yang dikelola di AWS Secrets Manager menggunakan CloudFormation templat AWS dan Menggunakan Referensi Dinamis untuk Menentukan Nilai Template.

- Gunakan snapshot Amazon RDS untuk membuat instans RDS DB — Dengan melakukan ini, Anda menghindari keharusan menyediakan. MasterUserPassword
- Jika template yang Anda kirimkan berisi profil instans IAM, itu harus diawali dengan 'pelanggan'. Misalnya, menggunakan profil instance dengan nama 'example-instance-profile', menyebabkan kegagalan. Sebagai gantinya, gunakan profil instance dengan nama 'customer-example-instance-profile'.
- Jangan sertakan data sensitif apa pun di **`AWS::EC2::Instance`** - [UserData]. UserData tidak boleh berisi kata sandi, kunci API, atau data sensitif lainnya. Jenis data ini dapat dienkripsi dan disimpan dalam bucket S3 dan diunduh ke instance menggunakan. UserData
- Pembuatan kebijakan IAM menggunakan CloudFormation templat didukung dengan kendala — kebijakan IAM harus ditinjau dan disetujui oleh AMS. SecOps Saat ini kami hanya mendukung penerapan peran IAM dengan kebijakan in-line yang berisi izin yang telah disetujui sebelumnya. Dalam kasus lain, kebijakan IAM tidak dapat dibuat menggunakan CloudFormation template karena itu akan menggantikan proses AMS SecOps .
- SSH KeyPairs tidak didukung - EC2 Instans Amazon harus diakses melalui sistem manajemen akses AMS. Proses AMS RFC mengautentikasi Anda. Anda tidak dapat menyertakan keypair SSH dalam CloudFormation template karena Anda tidak memiliki izin untuk membuat keypair SSH dan mengganti model manajemen akses AMS.
- Aturan masuknya Grup Keamanan dibatasi - Anda tidak dapat memiliki rentang CIDR sumber dari 0.0.0.0/0, atau ruang alamat yang dapat dirutekan secara publik, dengan port TCP yang tidak lain dari 80 atau 443.
- Ikuti CloudFormation panduan saat menulis templat CloudFormation sumber daya — Pastikan Anda menggunakan type/property nama data yang tepat untuk sumber daya dengan merujuk ke Panduan AWS CloudFormation Pengguna untuk sumber daya tersebut. Misalnya, tipe data SecurityGroupIds properti dalam AWS::EC2::Instance sumber daya adalah 'Daftar nilai String', jadi ["sg-aaaaaaaa"] ok (dengan tanda kurung), tetapi “sg-aaaaaaaa” tidak (tanpa tanda kurung).

  Untuk informasi selengkapnya, lihat AWS Resource and Property Types Reference.

- Konfigurasikan CloudFormation templat khusus Anda untuk menggunakan parameter yang ditentukan dalam CT CloudFormation cerna AMS — Saat mengonfigurasi CloudFormation templat untuk menggunakan parameter yang ditentukan dalam CT CloudFormation cerna AMS, Anda dapat menggunakan kembali CloudFormation templat untuk membuat tumpukan serupa dengan mengirimkannya dengan nilai parameter yang diubah dalam input CT dengan Management | Custom stack | Stack from CloudFormation template | Update CT (ct-361tlo1k7339x). Sebagai contoh, lihat CloudFormation Contoh menelan: Mendefinisikan sumber daya.
- Titik akhir bucket Amazon S3 dengan URL yang telah ditetapkan sebelumnya tidak dapat kedaluwarsa — Jika Anda menggunakan titik akhir bucket Amazon S3 dengan URL yang telah ditetapkan sebelumnya, verifikasi bahwa URL Amazon S3 yang telah ditetapkan sebelumnya tidak kedaluwarsa. RFC CloudFormation ingest yang dikirimkan dengan URL bucket Amazon S3 yang sudah kedaluwarsa ditolak.
- Kondisi Tunggu memerlukan logika sinyal - Kondisi Tunggu digunakan untuk mengoordinasikan pembuatan sumber daya tumpukan dengan tindakan konfigurasi yang berada di luar pembuatan tumpukan. Jika Anda menggunakan sumber daya Kondisi Tunggu di template, CloudFormation tunggu sinyal sukses, dan itu menandai pembuatan tumpukan sebagai kegagalan jika jumlah sinyal sukses tidak dibuat. Anda harus memiliki logika untuk sinyal jika Anda menggunakan sumber daya Kondisi Tunggu. Untuk informasi selengkapnya, lihat Membuat Kondisi Tunggu di Template.

Praktik terbaik

Berikut adalah beberapa praktik terbaik yang dapat Anda gunakan untuk memigrasikan sumber daya menggunakan proses CloudFormation konsumsi AMS:

- Kirimkan IAM dan sumber daya terkait kebijakan lainnya dalam satu CT - Jika Anda dapat menggunakan otomatis CTs seperti CloudFormation Ingest untuk menerapkan peran IAM, kami sarankan Anda melakukannya. Dalam kasus lain, AMS merekomendasikan agar Anda mengumpulkan semua IAM atau sumber daya terkait kebijakan lainnya dan mengirimkannya dalam satu Manajemen | Lainnya | Lainnya | Buat jenis perubahan (ct-1e1xtak34nx76). Misalnya, gabungkan semua peran IAM yang diperlukan, profil EC2 instans Amazon IAM, pembaruan kebijakan IAM untuk peran IAM yang ada, kebijakan bucket Amazon S3, kebijakan SNS/Amazon Amazon SQS, dan sebagainya, dan kirimkan ct-1e1xtak34nx76 RFC sehingga sumber daya yang sudah ada sebelumnya ini dapat dengan mudah direferensikan di dalam template ingest future. CloudFormation
- EC2 instance di-bootstrap dan berhasil bergabung ke domain — Ini dilakukan secara otomatis sebagai praktik terbaik. Untuk memastikan bahwa EC2 instans Amazon yang diluncurkan melalui

tumpukan CloudFormation ingest di-bootstrap dan berhasil bergabung dengan domain, AMS menyertakan sumber daya grup UpdatePolicy Auto Scaling CreationPolicy dan untuk (yaitu, jika kebijakan ini belum ada).

- Parameter instans Amazon RDS DB harus ditentukan — Saat membuat database Amazon RDS melalui CloudFormation ingest, Anda harus menentukan `DBSnapshotIdentifier` parameter untuk memulihkan dari snapshot DB sebelumnya. Ini diperlukan karena CloudFormation ingest saat ini tidak menangani data sensitif.

Untuk contoh cara menggunakan CloudFormation template untuk AMS CloudFormation template ingest, lihatCloudFormation Ingest: Contoh.

Validasi template

Anda dapat memvalidasi sendiri CloudFormation template Anda sebelum mengirimkannya ke AMS.

Template yang dikirimkan ke AMS CloudFormation ingest divalidasi untuk memastikan mereka aman untuk digunakan dalam akun AMS. Proses validasi memeriksa hal-hal berikut:

- Sumber daya yang didukung — Hanya sumber daya yang CloudFormation didukung AMS ingest yang digunakan. Untuk informasi selengkapnya, lihat Sumber Daya yang Didukung.
- Didukung AMIs - AMI dalam template adalah AMI yang didukung AMS. Untuk informasi tentang AMS AMIs, lihatGambar Mesin AMS Amazon (AMIs).
- Subnet AMS Shared Services — Template tidak mencoba meluncurkan sumber daya ke subnet AMS Shared Services.
- Kebijakan sumber daya — Tidak ada kebijakan sumber daya yang terlalu permisif, seperti kebijakan bucket S3 yang dapat dibaca publik atau dapat ditulis. AMS tidak mengizinkan bucket S3 yang dapat dibaca atau ditulis secara publik masuk. Akun AWS

Validasi dengan Linter CloudFormation

Anda dapat memvalidasi sendiri CloudFormation template Anda sebelum mengirimkannya ke AMS dengan menggunakan alat Linter. CloudFormation

Alat CloudFormation Linter adalah cara terbaik untuk memvalidasi CloudFormation template Anda karena menyediakan validasi untuk resource/property nama, tipe data, dan fungsi. Untuk informasi selengkapnya, lihat cfn-python-lintaws-cloudformation/.

Output CloudFormation Linter dari template yang ditunjukkan sebelumnya adalah sebagai berikut:

```
$ cfn-lint -t ./testtmpl.json
E3002 Invalid Property Resources/SNSTopic/Properties/Name
./testtmpl.json:6:9
```

Untuk membantu validasi offline CloudFormation template, AMS telah mengembangkan seperangkat aturan validasi kustom pluggable untuk alat Linter. CloudFormation Mereka berada di halaman Sumber Daya Pengembang di konsol AMS.

Ikuti langkah-langkah ini untuk menggunakan skrip CloudFormation validasi pra-konsumsi:

1. Instal alat CloudFormation Linter. Untuk petunjuk penginstalan, lihat aws-cloudformation/cfn-lint.
2. Unduh file.zip dengan skrip validasi:

   Aturan Kustom Lint CFN.
3. Buka zip aturan terlampir ke direktori pilihan Anda.
4. Validasi CloudFormation template Anda dengan menjalankan perintah berikut:

```
cfn-lint --template {TEMPLATE_FILE} --append-rules {DIRECTORY_WITH_CUSTOM_RULES}
```

CloudFormation tumpukan cerna: contoh validator CFN

Contoh-contoh ini dapat membantu Anda mempersiapkan template Anda untuk penyerapan yang sukses.

Format validasi

Validasi bahwa template berisi bagian “Sumber Daya”, dan semua sumber daya yang ditentukan di bawahnya memiliki nilai “Jenis”.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description" : "Create a SNS topic",
  "Resources": {
    "SnsTopic": {
      "Type": "AWS::SNS::Topic"
    }
  }
}
```

Validasi bahwa kunci root template diizinkan. Kunci root yang diizinkan adalah:

```
[
  "AWSTemplateFormatVersion",
  "Description",
  "Mappings",
  "Parameters",
  "Conditions",
  "Resources",
  "Rules",
  "Outputs",
  "Metadata"
]
```

Tinjauan manual diperlukan validasi

Jika template berisi sumber daya berikut, validasi otomatis gagal dan Anda memerlukan tinjauan manual.

Kebijakan yang ditampilkan adalah area berisiko tinggi dari sudut pandang keamanan. Misalnya, kebijakan bucket S3 yang mengizinkan siapa pun kecuali pengguna atau grup tertentu untuk membuat objek atau izin menulis, sangat berbahaya. Jadi kami memvalidasi kebijakan dan menyetujui atau menolak berdasarkan konten, dan kebijakan tersebut tidak dapat dibuat secara otomatis. Kami sedang menyelidiki kemungkinan pendekatan untuk mengatasi masalah ini.

Saat ini kami tidak memiliki validasi otomatis di sekitar sumber daya berikut.

```
[
    "S3::BucketPolicy",
    "SNS::TopicPolicy",
    "SQS::QueuePolicy"
]
```

Validasi parameter

Validasi bahwa jika parameter template tidak memiliki nilai yang disediakan; itu harus memiliki nilai default.

Validasi atribut sumber daya

Pemeriksaan atribut yang diperlukan: Atribut tertentu harus ada untuk jenis sumber daya tertentu.

- “VPCOptions” harus ada di `AWS::OpenSearch::Domain`
- “CludsterSubnetGroupName” harus ada di `AWS::Redshift::Cluster`

```
{
    "AWS::OpenSearch::Domain": [
      "VPCOptions"
    ],
    "AWS::Redshift::Cluster": [
      "ClusterSubnetGroupName"
    ]
}
```

Pemeriksaan atribut yang tidak diizinkan: Atribut tertentu harus*tidak* ada untuk jenis sumber daya tertentu.

- “SecretString" tidak boleh ada di "AWS::SecretsManager::Secret”
- “MongoDbSettings" tidak boleh ada di "AWS::DMS::Endpoint”

```
{
  "AWS::SecretsManager::Secret": [
    "SecretString"
  ],
  "AWS::DMS::Endpoint": [
    "MongoDbSettings"
  ]
}
```

Pemeriksaan parameter SSM: Untuk atribut dalam daftar berikut, nilai harus ditentukan melalui Secrets Manager atau Systems Manager Parameter Store (Parameter String Aman):

```
{
  "RDS::DBInstance": [
    "MasterUserPassword",
    "TdeCredentialPassword"
  ],
  "RDS::DBCluster": [
    "MasterUserPassword"
  ],
  "ElastiCache::ReplicationGroup": [
```

```
    "AuthToken"
  ],
  "DMS::Certificate": [
    "CertificatePem",
    "CertificateWallet"
  ],
  "DMS::Endpoint": [
    "Password"
  ],
  "CodePipeline::Webhook": {
    "AuthenticationConfiguration": [
        "SecretToken"
    ]
  },
  "DocDB::DBCluster": [
    "MasterUserPassword"
  ]
},
```

Beberapa atribut harus sesuai dengan pola tertentu; misalnya, nama profil instans IAM tidak boleh dimulai dengan awalan cadangan AMS, dan nilai atribut harus cocok dengan regex tertentu seperti yang ditunjukkan:

```
{
    "AWS::EC2::Instance": {
      "IamInstanceProfile": [
        "^(?!arn:aws:iam|ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|
sentinel|Sentinel).+",
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile/(?!ams|Ams|AMS|
AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    },
    "AWS::AutoScaling::LaunchConfiguration": {
      "IamInstanceProfile": [
        "^(?!arn:aws:iam|ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|
sentinel|Sentinel).+",
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile/(?!ams|Ams|AMS|
AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    },
    "AWS::EC2::LaunchTemplate": {
      "LaunchTemplateData.IamInstanceProfile.Name": [
```

```
        "^(?!ams|Ams|AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|
Sentinel).+"
      ],
      "LaunchTemplateData.IamInstanceProfile.Arn": [
        "arn:aws:iam::(\\$\\{AWS::AccountId\\}|[0-9]+):instance-profile\/(?!ams|Ams|
AMS|AWSManagedServices|Managed_Services|mc|Mc|MC|sentinel|Sentinel).+"
      ]
    }
}
```

Validasi sumber daya

Hanya sumber daya yang diizinkan yang dapat ditentukan dalam templat; sumber daya tersebut dijelaskan dalamSumber Daya yang Didukung.

EC2 tumpukan dan grup Auto Scaling ASGs () tidak diizinkan dalam tumpukan yang sama karena batasan penambalan.

Validasi aturan masuknya grup keamanan

- Untuk permintaan yang berasal dari CFN Ingest Create atau Stack Update CT jenis perubahan:
  - Jika (`IpProtocol`adalah tcp atau 6) AND (Port adalah 80 atau 443), tidak ada batasan di sekitar nilai `CidrIP`
  - Jika tidak, tidak `CidrIP` bisa 0.0.0.0/0
- Untuk permintaan yang berasal dari Service Catalog (produk Service Catalog):
  - Selain validasi jenis perubahan CFN Ingest Create atau Stack Update CT, port `management_ports` dengan protokol di hanya `ip_protocols` dapat diakses melalui: `allowed_cidrs`

    ```
    {
          "ip_protocols": ["tcp", "6", "udp", "17"],
          "management_ports": [22, 23, 389, 636, 1494, 1604, 2222, 3389, 5900, 5901,
     5985, 5986],
          "allowed_cidrs": ["10.0.0.0/8", "100.64.0.0/10", "172.16.0.0/12",
     "192.168.0.0/16"]
      }
    ```

Batasan

Fitur dan fungsionalitas berikut saat ini tidak didukung oleh proses CloudFormation konsumsi AMS.

- YALL - Tidak didukung. Hanya CloudFormation template berbasis JSON yang didukung.
- Tumpukan bersarang — Sebagai gantinya, arsiteksikan infrastruktur aplikasi Anda untuk menggunakan satu templat. Atau, sebagai alternatif Anda dapat menggunakan referensi cross-stack untuk memisahkan sumber daya di beberapa tumpukan di mana satu sumber daya memiliki ketergantungan pada yang lain. Untuk informasi selengkapnya, lihat Panduan: Lihat Output Sumber Daya di AWS Stack Lainnya. CloudFormation
- CloudFormation set tumpukan - Tidak didukung, karena implikasi keamanan.
- Pembuatan sumber daya IAM menggunakan CloudFormation templat - Hanya peran IAM yang didukung, karena implikasi keamanan.
- Data sensitif — Tidak didukung. Jangan sertakan data sensitif dalam template atau dalam nilai parameter. Jika Anda perlu mereferensikan data sensitif, gunakan Secrets Manager untuk menyimpan dan mengambil nilai-nilai ini. Untuk informasi tentang menggunakan rahasia AWS Secrets Manager di properti resource, lihat Cara membuat dan mengambil rahasia yang dikelola di AWS Secrets Manager menggunakan CloudFormation templat AWS dan Menggunakan Referensi Dinamis untuk Menentukan Nilai Templat.

Sumber Daya yang Didukung

Sumber daya AWS berikut didukung dalam proses CloudFormation konsumsi AMS.

CloudFormation Ingest Stack: Sumber daya yang didukung

Sistem operasi instance harus didukung oleh konsumsi beban kerja AMS. Hanya sumber daya AWS yang tercantum di sini yang didukung.

- Amazon API Gateway
  - AWS::ApiGateway::Account
  - AWS::ApiGateway::ApiKey
  - AWS::ApiGateway::Authorizer
  - AWS::ApiGateway::BasePathPemetaan
  - AWS::ApiGateway::ClientCertificate
  - AWS::ApiGateway::Deployment
  - AWS::ApiGateway::DocumentationPart
  - AWS::ApiGateway::DocumentationVersion

- AWS::ApiGateway::DomainName
- AWS::ApiGateway::GatewayResponse
- AWS::ApiGateway::Method
- AWS::ApiGateway::Model
- AWS::ApiGateway::RequestValidator
- AWS::ApiGateway::Resource
- AWS::ApiGateway::RestApi
- AWS::ApiGateway::Stage
- AWS::ApiGateway::UsagePlan
- AWS::ApiGateway::UsagePlanKunci
- AWS::ApiGateway::VpcLink

- Amazon API Gateway V2
  - AWS::ApiGatewayV2::Api
  - AWS::ApiGatewayV2::ApiGatewayManagedOverrides
  - AWS::ApiGatewayV2::ApiMapping
  - AWS::ApiGatewayV2::Authorizer
  - AWS::ApiGatewayV2::Deployment
  - AWS::ApiGatewayV2::DomainName
  - AWS::ApiGatewayV2::Integration
  - AWS::ApiGatewayV2::IntegrationResponse
  - AWS::ApiGatewayV2::Model
  - AWS::ApiGatewayV2::Route
  - AWS::ApiGatewayV2::RouteResponse
  - AWS::ApiGatewayV2::Stage
  - AWS::ApiGatewayV2::VpcLink
- AWS AppSync
  - AWS::AppSync::ApiCache
  - AWS::AppSync::ApiKey
  - AWS::AppSync::DataSource
  - AWS::AppSync::FunctionConfiguration

  - AWS::AppSync::GraphQLApi
  - AWS::AppSync::GraphQLSchema
  - AWS::AppSync::Resolver
- Amazon Athena
  - AWS::Athena::NamedQuery
  - AWS::Athena::WorkGroup
- AWS Backup
  - AWS::Backup::BackupVault
- Amazon CloudFront
  - AWS::CloudFront::Distribution
  - AWS::CloudFront::CloudFrontOriginAccessIdentity
  - AWS::CloudFront::StreamingDistribution
- Amazon CloudWatch
  - AWS::CloudWatch::Alarm
  - AWS::CloudWatch::AnomalyDetector
  - AWS::CloudWatch::CompositeAlarm
  - AWS::CloudWatch::Dashboard
  - AWS::CloudWatch::InsightRule
- CloudWatch Log Amazon
  - AWS::Logs::LogGroup
  - AWS::Logs::LogStream
  - AWS::Logs::MetricFilter
  - AWS::Logs::SubscriptionFilter
- Amazon Cognito
  - AWS::Cognito::IdentityPool
  - AWS::Cognito::IdentityPoolRoleAttachment
  - AWS::Cognito::UserPool
  - AWS::Cognito::UserPoolKlien
  - AWS::Cognito::UserPoolDomain

  - AWS::Cognito::UserPoolKelompok

  - AWS::Cognito::UserPoolIdentityProvider
  - AWS::Cognito::UserPoolResourceServer
  - AWS::Cognito::UserPoolRiskConfigurationAttachment
  - AWS::Cognito::UserPoolUICustomizationLampiran
  - AWS::Cognito::UserPoolPengguna
  - AWS::Cognito::UserPoolUserToGroupAttachment
- Amazon DocumentDB
  - AWS::DocDB:: DBCluster
  - AWS::DocDB:: DBCluster ParameterGroup
  - AWS::DocDB:: DBInstance
  - AWS::DocDB:: DBSubnet Grup
- Amazon DynamoDB
  - AWS::DynamoDB::Table
- Amazon EC2
  - AWS::EC2::Volume
  - AWS::EC2::VolumeAttachment
  - AWS::EC2::Instance
  - AWS::EC2: :EIP
  - AWS::EC2:: EIPAssociation
  - AWS::EC2::NetworkInterface
  - AWS::EC2::NetworkInterfaceLampiran
  - AWS::EC2::SecurityGroup
  - AWS::EC2::SecurityGroupMasuknya
  - AWS::EC2::SecurityGroupJalan keluar
  - AWS::EC2::LaunchTemplate
- AWS Batch
  - AWS::Batch::ComputeEnvironment
  - AWS::Batch::JobDefinition
  - AWS::Batch::JobQueue

- Registri Wadah Elastis Amazon (ECR)

  - AWS::ECR::Repository
- Layanan Kontainer Elastis Amazon (ECS) (Fargate)
  - AWS::ECS::CapacityProvider
  - AWS::ECS::Cluster
  - AWS::ECS::PrimaryTaskSet
  - AWS::ECS::Service
  - AWS::ECS::TaskDefinition
  - AWS::ECS::TaskSet
- Amazon Elastic File System (EFS)
  - AWS::EFS::FileSystem
  - AWS::EFS::MountTarget
- Amazon ElastiCache
  - AWS::ElastiCache::CacheCluster
  - AWS::ElastiCache::ParameterGroup
  - AWS::ElastiCache::ReplicationGroup
  - AWS::ElastiCache::SecurityGroup
  - AWS::ElastiCache::SecurityGroupMasuknya
  - AWS::ElastiCache::SubnetGroup
- Amazon EventBridge
  - AWS::Events::EventBus
  - AWS::Events::EventBusKebijakan
  - AWS::Events::Rule
- Amazon FSx
  - AWS::FSx::FileSystem
- Amazon Inspector
  - AWS::Inspector::AssessmentTarget
  - AWS::Inspector::AssessmentTemplate
  - AWS::Inspector::ResourceGroup
- Amazon Kinesis Data Analytics
  - AWS::KinesisAnalytics::Application

  - AWS::KinesisAnalytics::ApplicationOutput
  - AWS::KinesisAnalytics::ApplicationReferenceDataSource
- Amazon Kinesis Data Firehose
  - AWS::KinesisFirehose::DeliveryStream
- Amazon Kinesis Data Streams
  - AWS::Kinesis::Stream
  - AWS::Kinesis::StreamConsumer
- Amazon MQ
  - AWS::AmazonMQ::Broker
  - AWS::AmazonMQ::Configuration
  - AWS::AmazonMQ::ConfigurationAssociation
- Amazon OpenSearch
  - AWS::OpenSearchService::Domain
- Amazon Relational Database Service (RDS)
  - AWS: :RDS:: DBCluster
  - AWS: :RDS:: DBCluster ParameterGroup
  - AWS: :RDS:: DBInstance
  - AWS: :RDS:: Grup DBParameter
  - AWS: :RDS:: Grup DBSubnet
  - AWS::RDS::EventSubscription
  - AWS::RDS::OptionGroup
- Amazon Route 53
  - AWS::Route53::HealthCheck
  - AWS::Route53::HostedZone
  - AWS::Route53::RecordSet
  - AWS::Route53::RecordSetKelompok
  - AWS::Route53Resolver::ResolverRule
  - AWS::Route53Resolver::ResolverRuleAsosiasi
- Amazon S3
  - AWS::S3::Bucket

- Pembuat Sagemaker Amazon
  - AWS::SageMaker::CodeRepository
  - AWS::SageMaker::Endpoint
  - AWS::SageMaker::EndpointConfig
  - AWS::SageMaker::Model
  - AWS::SageMaker::NotebookInstance
  - AWS::SageMaker::NotebookInstanceLifecycleConfig
  - AWS::SageMaker::Workteam
- Amazon Simple Email Service (SES)
  - AWS::SES::ConfigurationSet
  - AWS::SES::ConfigurationSetEventDestination
  - AWS::SES::ReceiptFilter
  - AWS::SES::ReceiptRule
  - AWS::SES::ReceiptRuleSet
  - AWS::SES::Template
- Amazon SimpleDB
  - AWS::SDB::Domain
- Amazon SNS
  - AWS::SNS::Subscription
  - AWS::SNS::Topic
- Amazon SQS
  - AWS::SQS::Queue
- Amazon WorkSpaces
  - AWS::WorkSpaces::Workspace
- Aplikasi AutoScaling
  - AWS::ApplicationAutoScaling::ScalableTarget
  - AWS::ApplicationAutoScaling::ScalingPolicy
- Amazon EC2 AutoScaling
  - AWS::AutoScaling::AutoScalingKelompok
  - AWS::AutoScaling::LaunchConfiguration

  - AWS::AutoScaling::LifecycleHook
  - AWS::AutoScaling::ScalingPolicy
  - AWS::AutoScaling::ScheduledAction
- AWS Certificate Manager
  - AWS::CertificateManager::Certificate
- AWS CloudFormation
  - AWS::CloudFormation::CustomResource
  - AWS::CloudFormation::Designer
  - AWS::CloudFormation::WaitCondition
  - AWS::CloudFormation::WaitConditionMenangani
- AWS CodeBuild
  - AWS::CodeBuild::Project
  - AWS::CodeBuild::ReportGroup
  - AWS::CodeBuild::SourceCredential
- AWS CodeCommit
  - AWS::CodeCommit::Repository
- AWS CodeDeploy
  - AWS::CodeDeploy::Application
  - AWS::CodeDeploy::DeploymentConfig
  - AWS::CodeDeploy::DeploymentGroup
- AWS CodePipeline
  - AWS::CodePipeline::CustomActionTipe
  - AWS::CodePipeline::Pipeline
  - AWS::CodePipeline::Webhook
- AWS Database Migration Service (DMS)
  - AWS::DMS::Certificate
  - AWS::DMS::Endpoint
  - AWS::DMS::EventSubscription
  - AWS::DMS::ReplicationInstance

  - AWS::DMS::ReplicationSubnetKelompok

  - AWS::DMS::ReplicationTask

  MongoDbSettings Properti dalam AWS::DMS::Endpoint sumber daya tidak diperbolehkan.

  Properti berikut hanya diizinkan jika diselesaikan oleh AWS Secrets Manager: CertificatePem dan CertificateWallet properti di AWS::DMS::Certificate sumber daya, dan properti Kata Sandi di AWS::DMS::Endpoint sumber daya.
- AWS Elastic Load Balancing - Application Load Balancer/Network Load Balancer
  - AWS::ElasticLoadBalancingV2::Listener
  - AWS::ElasticLoadBalancingV2::ListenerCertificate
  - AWS::ElasticLoadBalancingV2::ListenerRule
  - AWS::ElasticLoadBalancingV2::LoadBalancer
  - AWS::ElasticLoadBalancingV2::TargetGroup
- AWS Elastic Load Balancing - Classic Load Balancer
  - AWS::ElasticLoadBalancing::LoadBalancer
- AWS Elemental MediaConvert
  - AWS::MediaConvert::JobTemplate
  - AWS::MediaConvert::Preset
  - AWS::MediaConvert::Queue
- AWS Elemental MediaStore
  - AWS::MediaStore::Container
- AWS Identity and Access Management (IAM)
  - AWS::IAM::Role
- AWS Managed Streaming for Apache Kafka (MSK)
  - AWS::MSK::Cluster
- AWS Glue
  - AWS::Glue::Classifier
  - AWS::Glue::Connection
  - AWS::Glue::Crawler
  - AWS::Glue::Database
  - AWS::Glue::DataCatalogEncryptionSettings
  - AWS::Glue::DevEndpoint

  - AWS::Glue::Job
  - AWS::Glue::MLTransform
  - AWS::Glue::Partition
  - AWS::Glue::SecurityConfiguration
  - AWS::Glue::Table
  - AWS::Glue::Trigger
  - AWS::Glue::Workflow
- AWS Key Management Service (KMS)
  - AWS::KMS::Key
  - AWS::KMS::Alias
- Formasi AWS Lake
  - AWS::LakeFormation::DataLakePengaturan
  - AWS::LakeFormation::Permissions
  - AWS::LakeFormation::Resource
- AWS Lambda
  - AWS::Lambda::Alias
  - AWS::Lambda::EventInvokeConfig
  - AWS::Lambda::EventSourcePemetaan
  - AWS::Lambda::Function
  - AWS::Lambda::LayerVersion
  - AWS::Lambda::LayerVersionIzin
  - AWS::Lambda::Permission
  - AWS::Lambda::Version
- Amazon Redshift
  - AWS::Redshift::Cluster
  - AWS::Redshift::ClusterParameterKelompok
  - AWS::Redshift::ClusterSubnetKelompok
- AWS Secrets Manager
  - AWS::SecretsManager::ResourcePolicy
  - AWS::SecretsManager::RotationSchedule

- AWS::SecretsManager::Secret
- AWS::SecretsManager::SecretTargetLampiran

- AWS Security Hub
  - AWS::SecurityHub::Hub
- AWS Step Functions
  - AWS::StepFunctions::Activity
  - AWS::StepFunctions::StateMachine
- AWS Systems Manager (SSM)
  - AWS::SSM::Parameter
- Amazon CloudWatch Synthetics
  - AWS::Synthetics::Canary
- AWS Transfer Family
  - AWS::Transfer::Server
  - AWS::Transfer::User
- AWS WAF
  - AWS::WAF::ByteMatchSet
  - AWS: :WAF:: IPSet
  - AWS::WAF::Rule
  - AWS::WAF::SizeConstraintSet
  - AWS::WAF::SqlInjectionMatchSet
  - AWS::WAF::WebACL
  - AWS::WAF::XssMatchSet
- AWS WAF Regional
  - AWS::WAFRegional::ByteMatchSet
  - AWS::WAFRegional::GeoMatchSet
  - AWS::WAFRegional:: IPSet
  - AWS::WAFRegional::RateBasedAturan
  - AWS::WAFRegional::RegexPatternSet
  - AWS::WAFRegional::Rule

- AWS::WAFRegional::SizeConstraintSet

  - AWS::WAFRegional::SqlInjectionMatchSet
  - AWS::WAFRegional::WebACL
  - AWS::WAFRegional::WebACLAssociation
  - AWS::WAFRegional::XssMatchSet
- AWS WAFv2
  - AWS::WAFv2:: IPSet
  - AWS::WAFv2::RegexPatternSet
  - AWS::WAFv2::RuleGroup
  - AWS::WAFv2::WebACL
  - AWS::WAFv2::WebACLAssociation

CloudFormation Ingest: Contoh

Temukan di sini beberapa contoh rinci tentang cara menggunakan tumpukan Buat dengan jenis perubahan CloudFormation template.

Untuk mengunduh satu set CloudFormation templat sampel per AWS Region, lihat Contoh Template.

Untuk informasi referensi tentang CloudFormation sumber daya, lihat Referensi AWS Resource dan Jenis Properti. Namun, AMS mendukung serangkaian sumber daya yang lebih kecil, yang dijelaskan dalamAMS CloudFormation menelan.

> **Note**
>
> AMS menyarankan Anda untuk mengumpulkan semua IAM atau sumber daya terkait kebijakan lainnya dan mengirimkannya dalam satu Manajemen | Lainnya | Lainnya | Buat jenis perubahan (ct-1e1xtak34nx76). Misalnya, gabungkan semua peran IAM yang diperlukan, profil instans IAM, pembaruan kebijakan IAM untuk peran IAM yang ada, kebijakan bucket S3, kebijakan, dan sebagainya, lalu kirimkan ct-1e1xtak34nx76 RFC sehingga sumber daya yang sudah ada sebelumnya ini dapat direferensikan di dalam template CFN Ingest masa depan. SNS/SQS

Topik

- CloudFormation Contoh menelan: Mendefinisikan sumber daya

- CloudFormation Contoh konsumsi: Aplikasi Web 3-tier

CloudFormation Contoh menelan: Mendefinisikan sumber daya

Saat menggunakan AMS CloudFormation ingest, Anda menyesuaikan CloudFormation template dan mengirimkannya ke AMS dalam RFC dengan tipe perubahan CloudFormation ingest (ct-36cn2avfrrj9v). Untuk membuat CloudFormation template yang dapat digunakan kembali beberapa kali, Anda menambahkan parameter konfigurasi tumpukan ke input eksekusi tipe perubahan CloudFormation ingest daripada hard coding mereka dalam template. CloudFormation Manfaat terbesar adalah Anda dapat menggunakan kembali template.

Skema input tipe perubahan CloudFormation konsumsi AMS memungkinkan Anda memilih hingga enam puluh parameter dalam CloudFormation templat dan memberikan nilai khusus.

Contoh ini menunjukkan cara mendefinisikan properti sumber daya, yang dapat digunakan dalam berbagai CloudFormation templat, sebagai parameter dalam CT CloudFormation menelan AMS. Contoh di bagian ini secara khusus menunjukkan penggunaan topik SNS.

Topik

- Contoh 1: Kode keras TopicName properti CloudFormation SNSTopic sumber daya
- Contoh 2: Gunakan SNSTopic sumber daya untuk mereferensikan parameter dalam tipe perubahan AMS
- Contoh 3: Buat topik SNS dengan mengirimkan file parameter eksekusi JSON dengan tipe perubahan konsumsi AMS
- Contoh 4: Kirim jenis perubahan baru yang mereferensikan CloudFormation template yang sama
- Contoh 5: Gunakan nilai parameter default dalam CloudFormation template

Contoh 1: Kode keras **`TopicName`** properti CloudFormation SNSTopic sumber daya

Dalam contoh ini, Anda membuat kode keras `TopicName` properti CloudFormation SNSTopic sumber daya dalam CloudFormation template. Perhatikan bahwa `Parameters` bagian tersebut kosong.

Untuk memiliki CloudFormation templat yang memungkinkan Anda mengubah nilai SNSTopic nama tumpukan baru tanpa harus membuat CloudFormation templat baru, Anda dapat menggunakan `Parameters` bagian AMS dari jenis perubahan CloudFormation ingest untuk membuat konfigurasi

itu. Dengan melakukan ini, Anda menggunakan CloudFormation template yang sama nanti untuk membuat tumpukan baru dengan `SNSTopic` nama yang berbeda.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "My SNS Topic",
  "Parameters" : {
  },
  "Resources" : {
    "SNSTopic" : {
      "Type" : "AWS::SNS::Topic",
      "Properties" : {
        "TopicName" : "MyTopicName"
      }
    }
  }
}
```

Contoh 2: Gunakan SNSTopic sumber daya untuk mereferensikan parameter dalam tipe perubahan AMS

Dalam contoh ini, Anda menggunakan `TopicName` properti `SNSTopic` resource yang ditentukan dalam CloudFormation template untuk mereferensikan tipe perubahan AMS. `Parameter`

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "My SNS Topic",
  "Parameters" : {
    "TopicName" : {
      "Type" : "String",
      "Description" : "Topic ID",
      "Default" : "MyTopicName"
    }
  },
  "Resources" : {
    "SNSTopic" : {
      "Type" : "AWS::SNS::Topic",
      "Properties" : {
        "TopicName" : { "Ref" : "TopicName"}
      }
    }
  }
```

```
}
```

Contoh 3: Buat topik SNS dengan mengirimkan file parameter eksekusi JSON dengan tipe perubahan konsumsi AMS

Dalam contoh ini, Anda mengirimkan file parameter eksekusi JSON dengan CT serapan AMS yang membuat topik SNS. `TopicName` Topik SNS harus didefinisikan dalam CloudFormation template dengan cara yang dapat dimodifikasi yang ditunjukkan dalam contoh ini.

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic1"}
  ],
  "TimeoutInMinutes": 60
}
```

Contoh 4: Kirim jenis perubahan baru yang mereferensikan CloudFormation template yang sama

Contoh JSON ini mengubah `TopicName` nilai SNS tanpa membuat perubahan pada CloudFormation template. Sebagai gantinya, Anda mengirimkan Deployment | Ingestion | Stack from CloudFormation Template | Buat jenis perubahan yang mereferensikan template CFN yang sama.

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic2"}
  ],
  "TimeoutInMinutes": 60
}
```

Contoh 5: Gunakan nilai parameter default dalam CloudFormation template

Dalam contoh ini, SNS `TopicName` = 'MyTopicName' dibuat karena tidak ada `TopicName` nilai yang diberikan dalam parameter `Parameters` eksekusi. Jika Anda tidak memberikan `Parameters` definisi, nilai parameter default dalam CloudFormation template digunakan.

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_PRESIGNED_URL",
  "VpcId": "VPC_ID",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "TimeoutInMinutes": 60
}
```

CloudFormation Contoh konsumsi: Aplikasi Web 3-tier

Menyerap CloudFormation template untuk Aplikasi Web 3-Tier standar.

Ini termasuk Application Load Balancer, grup target Application Load Balancer, grup Auto Scaling, template peluncuran grup Auto Scaling, Amazon Relational Database Service (RDS untuk SQL Server) dengan database MySQL, penyimpanan Parameter SSM, dan Secrets Manager. AWS AWS Biarkan 30-60 menit untuk berjalan melalui contoh ini.

Prasyarat

- Buat rahasia yang berisi nama pengguna dan kata sandi dengan nilai yang sesuai menggunakan AWS Secrets Manager. Anda dapat merujuk ke contoh template JSON ini (file zip) yang berisi nama rahasia`aams-shared/myapp/dev/dbsecrets`, dan menggantinya dengan nama rahasia Anda. Untuk informasi tentang menggunakan AWS Secrets Manager dengan AMS, lihatMenggunakan AWS Secrets Manager dengan sumber daya AMS.
- Siapkan parameter yang diperlukan di AWS SSM Parameter Store (PS). Dalam contoh ini, subnet Private dan Public disimpan di SSM PS di jalur seperti`/app/DemoApp/PublicSubnet1a`,, `PublicSubnet1cPrivateSubnet1a`, `PrivateSubnet1c` dan. `VPCId` `Subnet-Id` `VPCCidr` Perbarui jalur dan nama parameter dan nilai untuk kebutuhan Anda.
- Buat peran EC2 instans Amazon IAM dengan izin baca ke AWS Secrets Manager dan jalur Penyimpanan Parameter SSM (peran IAM yang dibuat dan digunakan dalam

contoh ini adalah). `customer-ec2_secrets_manager_instance_profile` Jika Anda membuat kebijakan standar IAM seperti peran profil instance, nama peran harus dimulai dengan. `customer-` Untuk membuat peran IAM baru, (Anda dapat menamainya`customer-ec2_secrets_manager_instance_profile`, atau yang lainnya) gunakan AMS change type Management | Applications | IAM instance profile | Create (ct-0ixp4ch2tiu04) CT, dan lampirkan kebijakan yang diperlukan. Anda dapat meninjau kebijakan standar AMS IAM, `customer_secrets_manager_policy` dan`customer_systemsmanager_parameterstore_policy`, di konsol AWS IAM untuk digunakan apa adanya atau sebagai referensi.

Menyerap CloudFormation template untuk aplikasi Web 3-Tier standar

1. Unggah contoh template CloudFormation JSON terlampir sebagai file zip, 3- tier-cfn-ingest .zip ke bucket S3 dan buat URL S3 yang ditandatangani untuk digunakan di CFN Ingest RFC. Untuk informasi selengkapnya, lihat presign. Template CFN juga bisa copy/pasted masuk ke CFN Ingest RFC saat Anda mengirimkan RFC melalui konsol AMS.
2. Buat RFC CloudFormation Ingest (Deployment | Ingestion | Stack from CloudFormation template | Create (ct-36cn2avfrrj9v)), baik melalui konsol AMS atau AMS CLI. Proses otomatisasi CloudFormation ingest memvalidasi CloudFormation template untuk memastikan bahwa template memiliki sumber daya yang didukung AMS yang valid, dan mematuhi standar keamanan.

   - Menggunakan konsol - Untuk jenis perubahan, pilih Deployment -> Ingestion -> Stack from CloudFormation Template -> Create, dan kemudian tambahkan parameter berikut sebagai contoh (perhatikan bahwa default untuk Multi AZDatabase adalah false):

     ```
     CloudFormationTemplateS3Endpoint: "https://s3-ap-
     southeast-2.amazonaws.com/amzn-s3-demo-bucket/3-tier-cfn-ingest.json?
     AWSAccessKeyId=#{S3_ACCESS_KEY_ID}&Expires=#{EXPIRE_DATE}&Signature=#{SIGNATURE}"
     VpcId: "VPC_ID"
     TimeoutInMinutes: 120
     IAMEC2InstanceProfile: "customer_ec2_secrets_manager_instance_profile"
     MultiAZDatabase: "true"
     WebServerCapacity: "2"
     ```

   - Menggunakan AWS CLI - Untuk detail tentang membuat RFCs menggunakan AWS CLI, lihat Membuat RFCs. Sebagai contoh, jalankan perintah berikut:

```
aws --profile=saml amscm create-rfc  --change-type-id ct-36cn2avfrrj9v
 --change-type-version "2.0" --title "TEST_CFN_INGEST" --execution-
parameters "{\"CloudFormationTemplateS3Endpoint\":\"https://s3-
ap-southeast-2.amazonaws.com/my-bucket/3-tier-cfn-ingest.json?
AWSAccessKeyId=#{S3_ACCESS_KEY_ID}&Expires=#{EXPIRE_DATE}&Signature=#{SIGNATURE}\",
\"TimeoutInMinutes\":120,\"Description\":\"TEST\",\"VpcId\":\"VPC_ID\",
\"Name\":\"MY_TEST\",\"Tags\":[{\"Key\":\"env\",\"Value\":\"test\"}],
\"Parameters\":[{\"Name\":\"IAMEC2InstanceProfile\",\"Value\":
\"customer_ec2_secrets_manager_instance_profile\"},{\"Name\":\"MultiAZDatabase\",
\"Value\":\"true\"},{\"Name\":\"VpcId\",\"Value\":\"VPC_ID\"},{\"Name\":
\"WebServerCapacity\",\"Value\":\"2\"}]}" --endpoint-url https://amscm.us-
east-1.amazonaws.com/operational/ --no-verify-ssl
```

Temukan URL Application Load Balancer di output eksekusi CloudFormation RFC untuk mengakses situs web. Untuk informasi tentang mengakses sumber daya, lihat Mengakses instans.

Buat CloudFormation tumpukan ingest

Membuat tumpukan CloudFormation ingest menggunakan konsol

Untuk membuat tumpukan CloudFormation ingest menggunakan konsol

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) di tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.

- Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.

3. Pada halaman Jalankan RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.

4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.

5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat tumpukan CloudFormation ingest menggunakan CLI

Untuk membuat tumpukan CloudFormation ingest menggunakan CLI

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode dijelaskan di sini.

2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

1. Siapkan CloudFormation template yang akan Anda gunakan untuk membuat tumpukan, dan unggah ke bucket S3 Anda. Untuk detail penting, lihat Panduan, Praktik Terbaik, dan Batasan AWS CloudFormation Ingest.
2. Buat dan kirimkan RFC ke AMS:

   - Buat dan simpan file JSON parameter eksekusi, sertakan parameter CloudFormation template yang Anda inginkan. Contoh berikut menamainya CreateCfnParams .json.

     Contoh tumpukan aplikasi Web CreateCfnParams file.json:

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "VpcId": "VPC_ID",
  "CloudFormationTemplateS3Endpoint": "$S3_URL",
  "TimeoutInMinutes": 120,
  "Tags": [
   {
    "Key":   "Enviroment Type"
    "Value": "Dev",
   },
   {
    "Key":   "Application"
    "Value": "PCS",
   }
  ],
  "Parameters": [
   {
    "Name": "Parameter-for-S3Bucket-Name",
    "Value":  "BUCKET-NAME"
   },
```

```
    {
     "Name": "Parameter-for-Image-Id",
     "Value":  "AMI-ID"
    }
   ],
}
```

Contoh file topik SNS CreateCfnParams .json:

```
{
  "Name": "cfn-ingest",
  "Description": "CFNIngest Web Application Stack",
  "CloudFormationTemplateS3Endpoint": "$S3_URL",
  "Tags": [
    {"Key": "Enviroment Type", "Value": "Dev"}
  ],
  "Parameters": [
    {"Name": "TopicName", "Value": "MyTopic1"}
  ]
}
```

3. Buat dan simpan file JSON parameter RFC dengan konten berikut. Contoh berikut menamainya CreateCfnRfc file.json:

```
{
   "ChangeTypeId": "ct-36cn2avfrrj9v",
   "ChangeTypeVersion": "2.0",
   "Title": "cfn-ingest"
}
```

4. Buat RFC, tentukan CreateCfnRfc file dan file: CreateCfnParams

```
aws amscm create-rfc --cli-input-json file://CreateCfnRfc.json  --execution-
parameters file://CreateCfnParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

> **Note**
>
> Jenis perubahan ini pada versi 2.0 dan otomatis (tidak dijalankan secara manual).
> Ini memungkinkan eksekusi CT berjalan lebih cepat, dan, parameter baru, CloudFormationTemplate, memungkinkan Anda untuk menempelkan CloudFormation template khusus ke RFC. Selain itu, Dalam versi ini, kami tidak melampirkan grup keamanan AMS default jika Anda menentukan grup keamanan Anda sendiri. Jika Anda tidak menentukan grup keamanan Anda sendiri dalam permintaan, AMS akan melampirkan grup keamanan default AMS. Di CFN Ingest v1.0, kami selalu menambahkan grup keamanan default AMS baik Anda menyediakan grup keamanan Anda sendiri atau tidak.
> AMS telah mengaktifkan 17 layanan AMS Self-Provisioned untuk digunakan dalam jenis perubahan ini. Untuk informasi tentang sumber daya yang didukung, lihat CloudFormation Ingest Stack: Sumber Daya yang Didukung.

> **Note**
>
> Versi 2.0 menerima endpoint S3 yang bukan URL presigned.
> Jika Anda menggunakan versi CT sebelumnya, nilai parameter CloudFormationTemplateS3Endpoint harus berupa URL yang telah ditentukan sebelumnya.
> Contoh perintah untuk menghasilkan URL bucket S3 yang telah ditetapkan sebelumnya (Mac/Linux):

```
export S3_PRESIGNED_URL=$(aws s3 presign DASHDASHexpires-in 86400
 s3://BUCKET_NAME/CFN_TEMPLATE.json)
```

> Contoh perintah untuk menghasilkan URL bucket S3 yang telah ditetapkan sebelumnya (Windows):

```
for /f %i in ('aws s3 presign DASHDASHexpires-in 86400
 s3://BUCKET_NAME/CFN_TEMPLATE.json') do set S3_PRESIGNED_URL=%i
```

> Lihat juga Membuat Bucket Amazon URLs S3 Pra-Tanda Tangan.

> **Note**
>
> Jika bucket S3 ada di akun AMS, Anda harus menggunakan kredensi AMS untuk perintah ini. Misalnya, Anda mungkin perlu menambahkan `--profile saml` setelah mendapatkan kredensi AMS AWS Security Token Service (AWS STS) Anda.

Jenis perubahan terkait:Menyetujui set perubahan CloudFormation tumpukan ingest, Perbarui CloudFormation tumpukan ingest

Untuk mempelajari AWS selengkapnya CloudFormation, lihat AWS CloudFormation. Untuk melihat CloudFormation template, buka AWS CloudFormation Template Reference.

Memvalidasi ingest CloudFormation

Template divalidasi untuk memastikan bahwa itu dapat dibuat di akun AMS. Jika melewati validasi, itu diperbarui untuk menyertakan sumber daya atau konfigurasi apa pun yang diperlukan agar sesuai dengan AMS. Ini termasuk menambahkan sumber daya seperti CloudWatch alarm Amazon untuk memungkinkan Operasi AMS memantau tumpukan.

RFC ditolak jika salah satu dari berikut ini benar:

- Sintaks RFC JSON salah atau tidak mengikuti format yang diberikan.
- URL presigned bucket S3 yang disediakan tidak valid.
- Template ini bukan CloudFormation sintaks yang valid.
- Template tidak memiliki default yang ditetapkan untuk semua nilai parameter.
- Template gagal validasi AMS. Untuk langkah-langkah validasi AMS, lihat informasi nanti dalam topik ini.

RFC gagal jika CloudFormation tumpukan gagal dibuat karena masalah pembuatan sumber daya.

Untuk mempelajari lebih lanjut tentang validasi dan validator CFN, lihat Validasi Template dan contoh CloudFormation ingest stack: CFN validator.

Perbarui CloudFormation tumpukan ingest

Memperbarui tumpukan CloudFormation ingest menggunakan konsol

Untuk memperbarui CloudFormation Ingest Stack menggunakan konsol

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) di tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Memperbarui tumpukan CloudFormation ingest menggunakan CLI

Untuk memperbarui tumpukan CloudFormation ingest menggunakan CLI

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode dijelaskan di sini.

2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

1. Siapkan CloudFormation template yang ingin Anda gunakan untuk memperbarui tumpukan, dan unggah ke bucket S3 Anda. Untuk detail penting, lihat Panduan, Praktik Terbaik, dan Batasan AWS CloudFormation Ingest.
2. Buat dan kirimkan RFC ke AMS:

   - Buat dan simpan file JSON parameter eksekusi, sertakan parameter CloudFormation template yang Anda inginkan. Contoh ini menamainya UpdateCfnParams .json.

     Contoh UpdateCfnParams file.json dengan pembaruan parameter sebaris:

     ```
     {
       "StackId": "stack-yjjoo9aicjyqw4ro2",
       "VpcId": "VPC_ID",
       "CloudFormationTemplate": "{\"AWSTemplateFormatVersion\":\"2010-09-09\",
     \"Description\":\"Create a SNS topic\",\"Parameters\":{\"TopicName\":{\"Type
     \":\"String\"},\"DisplayName\":{\"Type\":\"String\"}},\"Resources\":{\"SnsTopic
     \":{\"Type\":\"AWS::SNS::Topic\",\"Properties\":{\"TopicName\":{\"Ref\":
     \"TopicName\"},\"DisplayName\":{\"Ref\":\"DisplayName\"}}}}}",
       "TemplateParameters": [
     ```

```
    {
      "Key": "TopicName",
      "Value": "TopicNameCLI"
    },
    {
      "Key": "DisplayName",
      "Value": "DisplayNameCLI"
    }
  ],
  "TimeoutInMinutes": 1440
}
```

Contoh UpdateCfnParams file.json dengan titik akhir bucket S3 yang berisi template yang diperbarui: CloudFormation

```
{
  "StackId": "stack-yjjoo9aicjyqw4ro2",
  "VpcId": "VPC_ID",
  "CloudFormationTemplateS3Endpoint": "s3_url",
  "TemplateParameters": [
    {
      "Key": "TopicName",
      "Value": "TopicNameCLI"
    },
    {
      "Key": "DisplayName",
      "Value": "DisplayNameCLI"
    }
  ],
  "TimeoutInMinutes": 1080
}
```

3. Buat dan simpan file JSON parameter RFC dengan konten berikut. Contoh ini menamainya UpdateCfnRfc file.json.

```
{
   "ChangeTypeId": "ct-361tlo1k7339x",
   "ChangeTypeVersion": "1.0",
   "Title": "cfn-ingest-template-update"
}
```

4. Buat RFC, tentukan UpdateCfnRfc file dan file: UpdateCfnParams

```
aws amscm create-rfc --cli-input-json file://UpdateCfnRfc.json  --execution-
parameters file://UpdateCfnParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

- Jenis perubahan ini sekarang di versi 2.0. Perubahan termasuk menghapus AutoApproveUpdateForResourcesparameter, yang digunakan dalam versi 1.0 CT ini, dan menambahkan dua parameter baru: AutoApproveRiskyUpdatesdan BypassDriftCheck.
- Jika bucket S3 ada di akun AMS, Anda harus menggunakan kredensi AMS untuk perintah ini. Misalnya, Anda mungkin perlu menambahkan `--profile saml` setelah mendapatkan kredensi AMS AWS Security Token Service (AWS STS) Anda.
- Semua `Parameter` nilai untuk sumber daya dalam CloudFormation template harus memiliki nilai, baik melalui nilai default atau kustom melalui bagian parameter CT. Anda dapat mengganti nilai parameter dengan menyusun sumber daya CloudFormation template untuk mereferensikan kunci Parameter. Untuk contoh yang menunjukkan cara melakukannya, lihat contoh CloudFormation validator ingest stack: CFN.

  PENTING: Parameter yang hilang tidak diberikan secara eksplisit dalam formulir, default ke nilai yang saat ini ditetapkan pada tumpukan atau templat yang ada.
- Untuk daftar layanan yang disediakan sendiri yang dapat Anda tambahkan menggunakan CloudFormation Ingest, lihat CloudFormation Ingest Stack: Supported Resources.

  Untuk mempelajari selengkapnya CloudFormation, lihat AWS CloudFormation.

Memvalidasi ingest CloudFormation

Template divalidasi untuk memastikan bahwa itu dapat dibuat di akun AMS. Jika melewati validasi, itu diperbarui untuk menyertakan sumber daya atau konfigurasi apa pun yang diperlukan agar sesuai dengan AMS. Ini termasuk menambahkan sumber daya seperti CloudWatch alarm Amazon untuk memungkinkan Operasi AMS memantau tumpukan.

RFC ditolak jika salah satu dari berikut ini benar:

- Sintaks RFC JSON salah atau tidak mengikuti format yang diberikan.
- URL presigned bucket S3 yang disediakan tidak valid.
- Template ini bukan CloudFormation sintaks yang valid.
- Template tidak memiliki default yang ditetapkan untuk semua nilai parameter.
- Template gagal validasi AMS. Untuk langkah-langkah validasi AMS, lihat informasi nanti dalam topik ini.

RFC gagal jika CloudFormation tumpukan gagal dibuat karena masalah pembuatan sumber daya.

Untuk mempelajari lebih lanjut tentang validasi dan validator CFN, lihat Validasi Template dan contoh CloudFormation ingest stack: CFN validator.

Menyetujui set perubahan CloudFormation tumpukan ingest

Menyetujui dan memperbarui tumpukan CloudFormation ingest menggunakan konsol

Untuk menyetujui dan memperbarui tumpukan CloudFormation ingest menggunakan konsol

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) di tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.

4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Menyetujui dan memperbarui tumpukan CloudFormation ingest menggunakan CLI

Untuk menyetujui dan memperbarui tumpukan CloudFormation ingest menggunakan CLI

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

1. Keluarkan parameter eksekusi skema JSON untuk jenis perubahan ini ke file di folder Anda saat ini. Contoh ini menamainya CreateAsgParams .json:

```
aws amscm create-rfc --change-type-id "ct-1404e21baa2ox" --change-
type-version "1.0" --title "Approve Update" --execution-parameters
 file://PATH_TO_EXECUTION_PARAMETERS --profile saml
```

2. Ubah dan simpan skema sebagai berikut:

```
{
  "StackId": "STACK_ID",
  "VpcId": "VPC_ID",
  "ChangeSetName": "UPDATE-ef81e2bc-03f6-4b17-a3c7-feb700e78faa",
  "TimeoutInMinutes": 1080
}
```

Kiat

> **Note**
>
> Jika ada beberapa sumber daya dalam tumpukan, dan Anda hanya ingin menghapus sebagian dari sumber daya tumpukan, gunakan CloudFormation Update CT; lihat CloudFormation Ingest Stack: Update. Anda juga dapat mengirimkan kasus permintaan Layanan dan teknisi AMS dapat membantu Anda menyusun set perubahan, jika diperlukan.

Untuk mempelajari lebih lanjut tentang AWS CloudFormation, lihat AWS CloudFormation.

Perbarui CloudFormation perlindungan penghentian tumpukan

Memperbarui tumpukan perlindungan CloudFormation terminasi dengan konsol

Berikut ini menunjukkan jenis perubahan ini di konsol AMS.

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.

2. Pilih jenis perubahan populer (CT) di tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Memperbarui perlindungan penghentian CloudFormation tumpukan dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

Hanya tentukan parameter yang ingin Anda ubah. Parameter yang tidak ada mempertahankan nilai yang ada.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip escape saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc \
--change-type-id "ct-2uzbqr7x7mekd" \
--change-type-version "1.0" \
--title "Enable termination protection on CFN stack" \
--execution-parameters "{\"DocumentName\":\"AWSManagedServices-
ManageResourceTerminationProtection\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ResourceId\":[\"stack-psvnq6cupymio3enl\"],\"TerminationProtectionDesiredState\":
[\"enabled\"]}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON; contoh ini menamainya EnableTermPro CFNParams .json:

```
aws amscm get-change-type-version --change-type-id "ct-2uzbqr7x7mekd"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 EnableTermProCFNParams.json
```

2. Ubah dan simpan EnableTermPro CFNParams file, hanya mempertahankan parameter yang ingin Anda ubah. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
  "DocumentName": "AWSManagedServices-ManageResourceTerminationProtection",
  "Region": "us-east-1",
  "Parameters": {
    "ResourceId": ["stack-psvnq6cupymio3enl"],
    "TerminationProtectionDesiredState": ["enabled"]
  }
}
```

3. Keluarkan template RFC ke file di folder Anda saat ini; contoh ini menamainya EnableTermPro CFNRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > EnableTermProCFNRfc.json
```

4. Ubah dan simpan EnableTermPro CFNRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
    "ChangeTypeId": "ct-2uzbqr7x7mekd",
    "ChangeTypeVersion": "1.0",
    "Title": "Enable termination protection on CFN instance"
}
```

5. Buat RFC, tentukan EnableTermPro CFNRfc file dan file: EnableTermPro CFNParams

```
aws amscm create-rfc --cli-input-json file://EnableTermProCFNRfc.json  --execution-
parameters file://EnableTermProCFNParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

> **Note**
>
> Ada CT terkait untuk Amazon EC2, EC2 tumpukan: Memperbarui perlindungan penghentian.

Untuk mempelajari lebih lanjut tentang perlindungan penghentian, lihat Melindungi tumpukan agar tidak dihapus.

Penerapan IAM otomatis menggunakan ingest CFN atau pembaruan tumpukan di AMS CTs

Anda dapat menggunakan jenis perubahan AMS ini untuk menerapkan peran IAM (`AWS::IAM::Role`sumber daya) di landing zone multi-akun (MALZ) dan single-account landing zone (SALZ):

- Penerapan | Tertelan | Tumpukan dari CloudFormation Template | Buat (ct-36cn2avfrrj9v)
- Manajemen | Tumpukan Kustom | Tumpukan Dari CloudFormation Template | Pembaruan (ct-361tlo1k7339x)
- Manajemen | Tumpukan Kustom | Tumpukan Dari CloudFormation Template | Menyetujui dan Memperbarui (ct-1404e21baa2ox)

Validasi yang dilakukan pada peran IAM di template CFN Anda:

- ManagedPolicyArns: Atribut tidak ManagedPolicyArnsboleh ada di`AWS::IAM::Role`. Validasi melarang melampirkan kebijakan terkelola ke peran yang disediakan. Sebagai gantinya, izin untuk peran dapat dikelola menggunakan kebijakan sebaris melalui Kebijakan properti.
- PermissionsBoundary: Kebijakan yang digunakan untuk menetapkan batas izin untuk peran hanya dapat berupa kebijakan terkelola penjual AMS:. `AWSManagedServices_IAM_PermissionsBoundary` Kebijakan ini bertindak sebagai rel penjaga yang melindungi sumber daya infrastruktur AMS agar tidak dimodifikasi menggunakan peran yang disediakan. Dengan batas izin default ini, manfaat keamanan yang diberikan AMS dipertahankan.

  `AWSManagedServices_IAM_PermissionsBoundary`(default) diperlukan, tanpanya, permintaan ditolak.

- MaxSessionDuration: Durasi sesi maksimum yang dapat diatur untuk peran IAM adalah 1 hingga 4 jam. Standar teknis AMS memerlukan penerimaan risiko pelanggan untuk durasi sesi di atas 4 jam.
- RoleName: Ruang nama berikut dipertahankan oleh AMS dan tidak dapat digunakan sebagai awalan nama peran IAM:

```
AmazonSSMRole,
AMS,
Ams,
ams,
AWSManagedServices,
customer_developer_role,
customer-mc-,
Managed_Services,
MC,
Mc,
mc,
SENTINEL,
Sentinel,
sentinel,
StackSet-AMS,
StackSet-Ams,
StackSet-ams,
StackSet-AWS,
StackSet-MC,
StackSet-Mc,
StackSet-mc
```

- Kebijakan: Kebijakan inline yang disematkan dalam peran IAM hanya dapat menyertakan serangkaian tindakan IAM yang telah disetujui sebelumnya oleh AMS. Ini adalah batas atas dari semua tindakan IAM yang diizinkan untuk membuat peran IAM dengan (kebijakan kontrol). Kebijakan kontrol terdiri dari:
  - Semua tindakan dalam kebijakan AWS terkelola ReadOnlyAccess yang menyediakan akses hanya-baca ke semua Layanan AWS dan sumber daya
  - Tindakan berikut, dengan pembatasan tindakan S3 lintas akun yaitu tindakan S3 yang diizinkan hanya dapat dilakukan pada sumber daya yang ada di akun yang sama dengan peran yang sedang dibuat:

```
amscm:*,
amsskms:*,
lambda:InvokeFunction,
```

```
logs:CreateLogStream,
logs:PutLogEvents,
s3:AbortMultipartUpload,
s3:DeleteObject,
s3:DeleteObjectVersion,
s3:ObjectOwnerOverrideToBucketOwner,
s3:PutObject,
s3:ReplicateTags,
secretsmanager:GetRandomPassword,
sns:Publish
```

Setiap peran IAM yang dibuat atau diperbarui melalui CFN ingest dapat memungkinkan tindakan yang tercantum dalam kebijakan kontrol ini, atau tindakan yang dicakup dari (kurang permisif daripada) tindakan yang tercantum pada kebijakan kontrol. Saat ini kami mengizinkan tindakan IAM aman ini yang dapat dikategorikan sebagai tindakan hanya-baca, ditambah tindakan non-readonly yang disebutkan di atas yang tidak dapat dilakukan melalui CTs dan telah disetujui sebelumnya sesuai standar teknis AMS.

- AssumeRolePolicyDocument: Entitas berikut telah disetujui sebelumnya dan dapat dimasukkan dalam kebijakan kepercayaan untuk mengambil peran yang sedang dibuat:
  - Entitas IAM apa pun (peran, pengguna, pengguna root, sesi peran yang dianggap STS) di akun yang sama dapat mengambil peran tersebut.
  - Berikut ini Layanan AWS dapat mengambil peran:

```
apigateway.amazonaws.com,
autoscaling.amazonaws.com,
cloudformation.amazonaws.com,
codebuild.amazonaws.com,
codedeploy.amazonaws.com,
codepipeline.amazonaws.com,
datapipeline.amazonaws.com,
datasync.amazonaws.com,
dax.amazonaws.com,
dms.amazonaws.com,
ec2.amazonaws.com,
ecs-tasks.amazonaws.com,
ecs.application-autoscaling.amazonaws.com,
elasticmapreduce.amazonaws.com,
es.amazonaws.com,
events.amazonaws.com,
firehose.amazonaws.com,
```

```
glue.amazonaws.com,
lambda.amazonaws.com,
monitoring.rds.amazonaws.com,
pinpoint.amazonaws.com,
rds.amazonaws.com,
redshift.amazonaws.com,
s3.amazonaws.com,
sagemaker.amazonaws.com,
servicecatalog.amazonaws.com,
sns.amazonaws.com,
ssm.amazonaws.com,
states.amazonaws.com,
storagegateway.amazonaws.com,
transfer.amazonaws.com,
vmie.amazonaws.com
```

- Penyedia SAMP di akun yang sama dapat mengambil peran tersebut. Saat ini, satu-satunya nama penyedia SAMP yang didukung adalah`customer-saml`.

Jika satu atau lebih validasi gagal, RFC ditolak. Contoh alasan penolakan RFC terlihat seperti ini:

```
{"errorMessage":"[ 'LambdaRole: The maximum session duration (in seconds) should be a
 numeric value in the range 3600 to 14400 (i.e. 1 to 4 hours).', 'lambda-policy: Policy
 document is too permissive.']","errorType":"ClientError"}
```

Jika Anda memerlukan bantuan dengan validasi atau eksekusi RFC yang gagal, gunakan korespondensi RFC untuk menghubungi AMS. Untuk petunjuk, lihat Korespondensi dan lampiran RFC (konsol). Untuk pertanyaan lain, kirimkan permintaan layanan. Untuk mengetahui caranya, lihat Membuat Permintaan Layanan.

> **Note**
>
> Saat ini kami tidak menerapkan praktik terbaik IAM sebagai bagian dari validasi IAM kami. Untuk praktik terbaik IAM, lihat Praktik terbaik keamanan di IAM.

Membuat peran IAM dengan tindakan yang lebih permisif atau menegakkan praktik terbaik IAM

Buat entitas IAM Anda dengan jenis perubahan manual berikut:

- Deployment | Komponen tumpukan lanjutan | Identity and Access Management (IAM) | Buat entitas atau kebijakan (ct-3dpd8mdd9jn1r)
- Manajemen | Komponen tumpukan tingkat lanjut | Identity and Access Management (IAM) | Perbarui entitas atau kebijakan (ct-27tuth19k52b4)

Kami menyarankan Anda membaca dan memahami standar teknis kami sebelum mengajukan manual RFCs ini. Untuk akses, lihat Cara mengakses standar teknis.

> **Note**
>
> Setiap peran IAM yang dibuat langsung dengan jenis perubahan manual ini milik tumpukan individualnya sendiri dan tidak berada di tumpukan yang sama di mana sumber daya infrastruktur lainnya dibuat melalui CFN Ingest CT.

Memperbarui Peran IAM yang dibuat dengan konsumsi CFN melalui jenis perubahan manual ketika pembaruan tidak dapat dilakukan melalui jenis perubahan otomatis

Gunakan Manajemen | Komponen tumpukan lanjutan | Identity and Access Management (IAM) | Perbarui jenis perubahan entitas atau kebijakan (ct-27tuth19k52b4).

> **Important**
>
> Pembaruan pada peran IAM melalui CT manual tidak tercermin dalam templat tumpukan CFN dan menyebabkan penyimpangan tumpukan. Setelah peran diperbarui melalui permintaan manual ke status yang tidak lulus validasi kami, peran tidak dapat diperbarui lebih lanjut menggunakan Stack Update CT (ct-361tlo1k7339x) lagi selama itu terus tidak sesuai dengan validasi kami. Pembaruan CT hanya dapat digunakan jika template tumpukan CFN sesuai dengan validasi kami. Namun, tumpukan masih dapat diperbarui melalui Stack Update CT (ct-361tlo1k7339x), selama sumber daya IAM yang tidak sesuai dengan validasi kami tidak diperbarui dan template CFN melewati validasi kami.

Menghapus peran IAM Anda yang dibuat melalui ingest AWS CloudFormation

Jika Anda ingin menghapus seluruh tumpukan, gunakan jenis perubahan Delete Stack otomatis berikut. Untuk petunjuk, lihat Menghapus Tumpukan:

- Ubah Jenis ID: ct-0q0bic0ywqk6c
- Klasifikasi: Manajemen | Tumpukan standar | Tumpukan | Hapus dan Manajemen | Komponen tumpukan tingkat lanjut | Tumpukan | Hapus

Jika Anda ingin menghapus peran IAM tanpa menghapus seluruh tumpukan, Anda dapat menghapus peran IAM dari CloudFormation template dan menggunakan template yang diperbarui sebagai input ke jenis perubahan Pembaruan Tumpukan otomatis:

- Ubah Tipe ID: ct-361tlo1k7339x
- Klasifikasi: Manajemen | Tumpukan khusus | Tumpukan dari CloudFormation templat | Perbarui

Untuk petunjuknya, lihat Memperbarui tumpukan AWS CloudFormation ingest.

CodeDeploy permintaan

Anda dapat menggunakan AWS CodeDeploy untuk membuat wadah aplikasi yang kemudian dapat Anda gunakan melalui grup CodeDeploy aplikasi. Untuk informasi selengkapnya CodeDeploy, lihat AWS CodeDeploy Documentation.

Bekerja dengan AWS CodeDeploy melibatkan proses berikut:

1. Buat CodeDeploy aplikasi. CodeDeploy Aplikasi adalah nama atau wadah yang digunakan oleh CodeDeploy untuk memastikan bahwa revisi yang benar, konfigurasi penyebaran, dan grup penyebaran direferensikan selama penerapan.
2. Buat grup CodeDeploy penyebaran. Grup CodeDeploy penyebaran mendefinisikan satu set instance individual yang ditargetkan untuk penerapan. AMS memiliki jenis perubahan terpisah untuk grup CodeDeploy penerapan untuk EC2.
3. Menyebarkan CodeDeploy aplikasi melalui grup CodeDeploy penyebaran.

CodeDeploy aplikasi

Membuat atau menyebarkan CodeDeploy aplikasi.

Buat CodeDeploy aplikasi

Membuat CodeDeploy aplikasi dengan konsol

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi yang lebih lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat CodeDeploy aplikasi dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk

parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.

2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip escape saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc --change-type-id "ct-0ah3gwb9seqk2" --change-type-version "1.0"
 --title "Stack-Create-CD-App" --execution-parameters "{\"Description\":\"TestCdApp\",
\"VpcId\":\"VPC_ID\",\"StackTemplateId\":\"stm-sft6rv00000000000\",\"Name\":\"Test\",
\"TimeoutInMinutes\":60,\"Parameters\":{\"CodeDeployApplicationName\":\"Test\"}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi skema JSON untuk CodeDeploy aplikasi CT ke file di folder Anda saat ini; contoh ini menamainya Create CDApp Params.json:

```
aws amscm get-change-type-version --change-type-id "ct-0ah3gwb9seqk2" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateCDAppParams.json
```

2. Ubah dan simpan file JSON sebagai berikut. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"Description":                      "Create WP CodeDeploy App",
"VpcId":                            "VPC_ID",
"StackTemplateId":                  "stm-sft6rv00000000000",
"Name":                             "WpCDApp",
"TimeoutInMinutes":                 60,
"Parameters":   {
    "CodeDeployApplicationName":    "WordPressCDApp"
    }
}
```

3. Keluarkan template JSON CreateRfc ke file di folder Anda saat ini; contoh ini menamainya Create CDApp RFC.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDAppRfc.json
```

4. Ubah dan simpan file JSON sebagai berikut. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0ah3gwb9seqk2",
"Title":                "CD-App-Stack-RFC"
}
```

5. Buat RFC, tentukan file Create CDApp Rfc dan file parameter eksekusi:

```
aws amscm create-rfc --cli-input-json file://CreateCDAppRfc.json --execution-
parameters file://CreateCDAppParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Untuk informasi selengkapnya tentang AWS CodeDeploy, lihat Membuat Aplikasi dengan AWS CodeDeploy.

Menyebarkan aplikasi CodeDeploy

Menyebarkan CodeDeploy aplikasi dengan konsol

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi yang lebih lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.

5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Menyebarkan CodeDeploy aplikasi dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip escape saat memberikan parameter eksekusi sebaris) dan kemudian kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc --change-type-id "ct-2edc3sd1sqmrb" --change-
type-version "2.0" --title "Stack-Deploy-CD-App" --execution-
parameters "{\"Description\":\"MyCDAppDeployTest\",\"VpcId\":
\"VPC_ID\",\"Name\":\"Test\",\"TimeoutInMinutes\":60,\"Parameters\":
{\"CodeDeployApplicationName\":\"TestCDApp\",\"CodeDeployDeploymentConfigName\":
\"CodeDeployDefault.OneAtATime\",\"CodeDeployDeploymentGroupName\":\"TestCDDepGroup\",
\"CodeDeployIgnoreApplicationStopFailures\":false,\"CodeDeployRevision\":
{\"RevisionType\":\"S3\",\"S3Location\":{\"S3Bucket\":\"amzn-s3-demo-bucket\",
\"S3BundleType\":\"tar\",\"S3Key\":\"TestKey\"}}}}"Test\"}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi skema JSON untuk CT CodeDeploy penerapan aplikasi; contoh ini menamainya Deploy Params.json: CDApp

   ```
   aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > DeployCDAppParams.json
   ```

2. Ubah file JSON sebagai berikut. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   "Description":                      "Deploy WordPress CodeDeploy Application",
   "VpcId":                            "VPC_ID",
   "Name":                             "WP CodeDeploy Deployment Group",
   "TimeoutInMinutes":                 360,
   "Parameters":   {
       "CodeDeployApplicationName":        "WordPressCDApp",
       "CodeDeployDeploymentGroupName":    "WordPressCDDepGroup",
       "CodeDeployIgnoreApplicationStopFailures": false,
       "CodeDeployRevision": {
         "RevisionType": "S3",
         "S3Location": {
           "S3Bucket": "amzn-s3-demo-bucket",
           "S3BundleType": "zip",
           "S3Key": "wordpress.zip" }
           }
       }
   }
   ```

3. Keluarkan template JSON CreateRfc ke file di folder Anda saat ini; contoh ini menamainya Deploy CDApp RFC.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

4. Ubah dan simpan file Deploy CDApp RFC.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "2.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-CD-APP-Stack-RFC"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file Deploy CDApp Rfc:

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

   Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Untuk informasi selengkapnya, lihat Membuat penerapan dengan CodeDeploy.

CodeDeploy grup penyebaran

Buat grup CodeDeploy aplikasi.

Buat CodeDeploy grup penyebaran

Membuat grup CodeDeploy penyebaran dengan konsol

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.

- Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

  Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi yang lebih lama muncul di sebelah tombol Buat RFC.
- Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.

3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat grup CodeDeploy penerapan dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip escape saat memberikan parameter eksekusi sebaris) dan kemudian kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc --change-type-id "ct-2gd0u847qd9d2" --change-type-version
 "1.0" --title "Stack-Create-CD-Dep-Group" --execution-parameters "{\"Description
\":\"TestCdDepGroupRfc\",\"VpcId\":\"VPC_ID\",\"StackTemplateId\":\"stm-
sp9lrk00000000000\",\"Name\":\"MyTestCDDepGroup\",\"TimeoutInMinutes\":60,\"Parameters
\":{\"CodeDeployApplicationName\":\"TestCDApp\",\"CodeDeployAutoScalingGroups\":
[\"TestASG\"],\"CodeDeployDeploymentConfigName\":\"CodeDeployDefault.OneAtATime\",
\"CodeDeployDeploymentGroupName\":\"Test\",\"CodeDeployServiceRoleArn\":
\"arn:aws:iam::000000000:role/aws-codedeploy-role\"}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi skema JSON ke file di folder Anda saat ini; contoh ini menamainya Create CDDep GroupParams .json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-2gd0u847qd9d2"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateCDDepGroupParams.json
   ```

2. Ubah dan simpan file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"Description":                      "CreateCDDeploymentGroup",
"VpcId":                            "VPC_ID",
"StackTemplateId":                  "stm-sp9lrk00000000000",
"Name":                             "WordPressCDAppGroup",
"TimeoutInMinutes":                 60,
"Parameters":   {
    "CodeDeployApplicationName":        "WordPressCDApp",
    "CodeDeployAutoScalingGroups":      ["ASG_NAME"],
    "CodeDeployDeploymentConfigName":   "CodeDeployDefault.HalfAtATime",
    "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
    "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
codedeploy-role"
    }
}
```

3. Keluarkan template JSON CreateRfc ke file di folder Anda saat ini; contoh ini menamainya Create CDDep GroupRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupRfc.json
```

4. Ubah dan simpan file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2gd0u847qd9d2",
"Title":                "CD-Dep-Group-RFC"
}
```

5. Buat RFC, tentukan CDDep GroupRfc file Buat dan file parameter eksekusi:

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupRfc.json --execution-
parameters file://CreateCDDepGroupParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Untuk informasi selengkapnya tentang grup CodeDeploy penerapan AWS, lihat Membuat Grup Penerapan dengan AWS. CodeDeploy

Buat grup CodeDeploy penyebaran untuk EC2

Membuat grup CodeDeploy penerapan untuk EC2 dengan konsol

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi yang lebih lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.

5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat grup CodeDeploy penerapan untuk EC2 dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip escape saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc --change-type-id "ct-00tlkda4242x7" --change-type-
version "1.0" --title "Stack-Create-CD-Ec2-Dep-Group" --execution-parameters
 "{\"Description\":\"MyTestCdDepEc2DepGroup\",\"VpcId\":\"VPC_ID\",\"Name\":
\"TestCDDepEc2Group\",\"StackTemplateId\":\"stm-n3hsoirgqeqqdbpk2\",\"TimeoutInMinutes
\":60,\"Parameters\":{\"ApplicationName\":\"TestCDApp\",\"DeploymentConfigName\":
\"CodeDeployDefault.OneAtATime\",\"AutoRollbackEnabled\":\"False\",\"EC2FilterTag\":
\"Name=Test\",\"EC2FilterTag2\":\"\",\"EC2FilterTag3\":\"\",\"ServiceRoleArn\":\"\"}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi skema JSON ke file; contoh ini menamainya Create CDDep GroupEc 2Params.json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-00tlkda4242x7"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateCDDepGroupEc2Params.json
   ```

2. Ubah dan simpan file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   "Description":                    "CreateCDDepGroupEc2",
   "VpcId":                          "VPC_ID",
   "StackTemplateId":                "stm-n3hsoirgqeqqdbpk2",
   "Name":                           "CDAppGroupEc2",
   "TimeoutInMinutes":               60,
   "Parameters":   {
       "ApplicationName":        "CDAppEc2",
       "DeploymentConfigName":   "CodeDeployDefault.OneAtATime",
       "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
       "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
   codedeploy-role"
       }
   }
   ```

3. Keluarkan template JSON CreateRfc ke file di folder Anda saat ini; contoh ini menamainya Create CDDep GroupEc 2RFC.json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupEc2Rfc.json
   ```

4. Ubah dan simpan file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   ```

```
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-00tlkda4242x7",
"Title":                "CD-Dep-Group-For-Ec2-Stack-RFC"
}
```

5. Buat RFC, tentukan file Create CDDep GroupEc 2Rfc dan file parameter eksekusi:

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupEc2Rfc.json --
execution-parameters file://CreateCDDepGroupEc2Params.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Untuk informasi selengkapnya tentang grup CodeDeploy penerapan AWS, lihat Membuat Grup Penerapan dengan AWS. CodeDeploy

AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS) membantu Anda memigrasikan database ke AMS dengan mudah dan aman. Anda dapat memigrasikan data Anda ke dan dari basis data komersial dan sumber terbuka yang paling banyak digunakan, seperti Oracle, MySQL, dan PostgreSQL. Layanan ini mendukung migrasi homogen seperti Oracle ke Oracle, dan juga migrasi heterogen antara platform database yang berbeda, seperti Oracle ke PostgreSQL atau MySQL ke Oracle. AWS DMS adalah AWS layanan; AMS CTs membantu Anda membuat AWS DMS sumber daya di akun yang dikelola AMS

Grafik berikut menggambarkan alur kerja migrasi database.

Topik

- AWS Database Migration Service (AWS DMS), sebelum Anda mulai
- AWS DMS, data yang diperlukan untuk penyiapan
- AWS DMS tugas pengaturan
- AWS DMS manajemen

AWS Database Migration Service (AWS DMS), sebelum Anda mulai

Saat merencanakan migrasi database menggunakan AMS AWS DMS, pertimbangkan hal berikut:

- Titik akhir sumber dan target: Anda perlu mengetahui informasi dan tabel apa dalam database sumber yang perlu dimigrasi ke database target. AMS AWS DMS mendukung migrasi skema dasar, termasuk pembuatan tabel dan kunci utama. Namun, AMS AWS DMS tidak secara otomatis membuat indeks sekunder, kunci asing, akun, dan sebagainya dalam database target. Lihat Sumber untuk Migrasi Data dan Target untuk Migrasi Data untuk informasi selengkapnya.
- Migrasi Skema/Kode: AMS AWS DMS tidak melakukan konversi skema atau kode. Anda dapat menggunakan alat seperti Oracle SQL Developer, MySQL Workbench, atau pgAdmin III untuk mengonversi skema Anda. Jika Anda ingin mengonversi skema yang ada ke mesin database yang berbeda, Anda dapat menggunakan AWS Schema Conversion Tool. Hal ini dapat membuat skema target dan juga dapat menghasilkan dan membuat seluruh skema: tabel, indeks, tampilan, dan sebagainya. Anda juga dapat menggunakan alat untuk mengonversi PL/SQL atau TSQL ke PGSQL dan format lainnya.
- Tipe data yang tidak didukung: Beberapa tipe data sumber perlu diubah menjadi tipe data yang setara untuk database target.

AWS DMS skenario untuk dipertimbangkan

Skenario berikut, didokumentasikan, dapat membantu Anda membuat jalur migrasi database Anda sendiri.

- Memigrasikan data dari server MySQL lokal ke Amazon RDS MySQL: Lihat postingan blog AWS Memigrasikan Data MySQL Lokal ke Amazon RDS (dan kembali)
- Memigrasikan data dari database Oracle ke Amazon RDS Aurora PostgreSQL database: Lihat postingan blog AWS Pengantar singkat untuk bermigrasi dari database Oracle ke database Amazon Aurora PostgreSQL
- Migrasi data dari RDS MySQL ke S3: Lihat postingan blog AWS Cara mengarsipkan data dari database relasional ke Amazon Glacier menggunakan AWS DMS

Untuk migrasi database, Anda harus melakukan hal berikut:

- Rencanakan migrasi database Anda, ini termasuk menyiapkan grup subnet replikasi.
- Alokasikan instance replikasi yang melakukan semua proses untuk migrasi.

- Tentukan sumber dan titik akhir basis data target.
- Buat tugas atau serangkaian tugas untuk menentukan tabel dan proses replikasi apa yang ingin Anda gunakan.
- Buat AWS DMS IAM `dms-cloudwatch-logs-role` dan `dms-vpc-role` peran. Jika Anda menggunakan Amazon Redshift sebagai database target, Anda juga harus membuat dan menambahkan peran IAM ke akun AWS `dms-access-for-endpoint` Anda. Untuk informasi selengkapnya, lihat Membuat peran IAM untuk digunakan dengan AWS CLI dan AWS DMS API.

Panduan ini memberikan contoh penggunaan konsol AMS atau AMS CLI untuk membuat (). AWS Database Migration Service AWS DMS Perintah CLI untuk membuat instance AWS DMS replikasi, grup subnet, dan tugas serta titik akhir AWS DMS sumber dan titik akhir target disediakan.

Untuk mempelajari lebih lanjut tentang AMS AWS DMS, lihat AWS Database Migration Serviceinformasi umum dan jawaban AWS Database Migration Service FAQsatas pertanyaan umum.

AWS DMS, data yang diperlukan untuk penyiapan

Untuk setiap AWS DMS penelusuran berikut, beberapa data yang sama diperlukan.

- `Description`: Informasi yang berarti tentang sumber daya, ini terpisah dari `Description` opsi parameter lainnya.
- `VpcId`: VPC yang akan digunakan. Anda dapat mengetahuinya dengan menjalankan ListVpcSummaries pengoperasian API SKMS (`list-vpc-summaries`di CLI) atau dengan melihat halaman di VPCsKonsol AMS. Untuk referensi AMS SKMS API, lihat tab Laporan di AWS Artifact Console.
- `Name`: Nama untuk komponen tumpukan atau tumpukan; ini menjadi Nama Stack.
- `TimeoutInMinutes`: Berapa menit yang diizinkan untuk pembuatan tumpukan sebelum RFC gagal. Pengaturan ini tidak akan menunda eksekusi RFC, tetapi Anda harus memberikan waktu yang cukup (misalnya, jangan tentukan`"5"`).
- `ChangeTypeId`,`ChangeTypeVersion`, dan`StackTemplateId`: Ini diperlukan tetapi bervariasi per CT dan nilainya disediakan di setiap bagian yang relevan, berikut.

AWS DMS tugas pengaturan

Siapkan AWS DMS dengan panduan berikut.

1: grup subnet AWS DMS replikasi: Buat

Anda dapat menggunakan konsol AMS atau API/CLI membuat grup subnet AWS DMS replikasi AMS.

Buat grup AWS DMS subnet replikasi

Membuat grup subnet AWS DMS replikasi dengan konsol

> **Note**
>
> CT ini gagal jika peran `dms-vpc-role` IAM tidak ada di akun.

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.

4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat grup subnet AWS DMS replikasi dengan CLI

> **Note**
>
> CT ini gagal jika peran `dms-vpc-role` IAM tidak ada di akun.

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke

bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris) dan kemudian kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-2q5azjd8p1ag5" --change-type-version "1.0" --title "TestDMSRepSG" --execution-
parameters "{\"Description\":\"DMSTestRepSG\",\"VpcId\":\"VPC-ID\",\"Name\":\"Test
 Stack\",\"Parameters\":{\"Description\":\"DESCRIPTION\",\"SubnetIds\":[\"SUBNET-ID\",
 \"SUBNET-ID\"]},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-j637f96ls1h4oy5fj
\"}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON; contoh ini menamainya CreateDmsRsgParams .json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-2q5azjd8p1ag5" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRsgParams.json
   ```

2. Ubah dan simpan parameter eksekusi CreateDmsRsgParams file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   "Description":          "DMSTestRepSG",
   "VpcId":                "VPC_ID",
   "TimeoutInMinutes":     60,
   "StackTemplateId":      "stm-j637f96ls1h4oy5fj",
   "Name":                 "Test RSG",
   "Parameters":   {
       "Description":            "DESCRIPTION",
       "SubnetIds":              ["SUBNET_ID", "SUBNET_ID"]
       }
   }
   ```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsRsgRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsRsgRfc.json
```

4. Ubah dan simpan CreateDmsRsgRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2q5azjd8p1ag5",
"Title":                "DMS-RSG-Create-RFC"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: CreateDmsRsgRfc

```
aws amscm create-rfc --cli-input-json file://CreateDmsRsgRfc.json --execution-
parameters file://CreateDmsRsgParams.json
```

   Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

- CT ini gagal jika peran `dms-vpc-role` IAM tidak ada di akun.
- Anda dapat menambahkan hingga 50 tag, tetapi untuk melakukannya Anda harus mengaktifkan tampilan konfigurasi tambahan.

  Untuk informasi selengkapnya tentang instans replikasi DMS dan grup subnet, lihat Menyiapkan Jaringan untuk Instance Replikasi.

2: contoh AWS DMS replikasi: Buat

Anda dapat menggunakan konsol AMS atau API/CLI membuat instance AWS DMS replikasi AMS.

Buat AWS DMS contoh replikasi

Membuat instance AWS DMS replikasi dengan konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat instance AWS DMS replikasi dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-27apldkhqr0ol" --change-type-version "1.0" --title "TestDMSRepInstance" --
execution-parameters "{\"Description\":\"DMSTestRepInstance\",\"VpcId\":\"VPC-ID\",
\"Name\":\"REP-INSTANCE-NAME\",\"Parameters\":{\"InstanceClass\":\"dms.t2.micro\",
```

```
\"ReplicationSubnetGroupIdentifier\":\"TEST-REP-SG\",\"SecurityGroupIds\":\"SG-ID, SG-ID\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-3n1j5hdrmiiiuqk6v\"}"
```

Saat instance replikasi Anda sedang dibuat, Anda dapat menentukan sumber dan penyimpanan data target. Penyimpanan data sumber dan target dapat dilakukan di instans Amazon Elastic Compute Cloud (Amazon EC2), Bucket AWS S3, instans DB Amazon Relational Database Service (Amazon RDS), atau database lokal.

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON; contoh ini menamainya CreateDmsRiParams .json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-27apldkhqr0ol" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRiParams.json
   ```

2. Ubah dan simpan parameter eksekusi CreateDmsRiParams file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   "Description":          "DMSTestRepInstance",
   "VpcId":                "VPC_ID",
   "Name":                 "Test RI",
   "StackTemplateId":      "stm-3n1j5hdrmiiiuqk6v",
   "TimeoutInMinutes":     60,
   "Parameters":   {
       "Description":                      "DESCRIPTION",
       "InstanceClass":                    "dms.t2.micro",
       "ReplicationSubnetGroupIdentifier": "TEST-REP-SG",
       "SecurityGroupIds":                 ["SG-ID, SG-ID"}
       }
   }
   ```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsRiRfc .json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > CreateDmsRiRfc.json
   ```

4. Ubah dan simpan CreateDmsRiRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-27apldkhqr0ol",
"Title":                "DMS-RI-Create-RFC"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: CreateDmsRiRfc

```
aws amscm create-rfc --cli-input-json file://CreateDmsRiRfc.json --execution-
parameters file://CreateDmsRiParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

- Anda dapat menambahkan hingga 50 tag, tetapi untuk melakukannya Anda harus mengaktifkan tampilan konfigurasi tambahan.
- Anda harus membuat instance replikasi pada instans di VPC AMS yang memiliki daya penyimpanan dan pemrosesan yang memadai untuk melakukan tugas yang Anda tetapkan dan memigrasikan data dari database sumber ke database target. EC2 Ukuran yang diperlukan dari instans ini bervariasi tergantung pada jumlah data yang perlu Anda migrasi dan tugas yang perlu dilakukan instans tersebut. Instans replikasi menyediakan ketersediaan tinggi dan dukungan failover menggunakan penerapan Multi-AZ saat Anda memilih opsi. `MultiAZ` Untuk informasi selengkapnya tentang instans replikasi, lihat Bekerja dengan Instans Replikasi AWS DMS.

3: titik akhir AWS DMS sumber: Buat, buat untuk Mongo DB, buat untuk S3

Anda dapat menggunakan konsol AMS atau API/CLI untuk membuat titik akhir sumber AMS DMS untuk berbagai database, kami menyediakan tiga contoh.

Titik akhir sumber DMS: membuat

Membuat Endpoint Sumber DMS dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat Endpoint Sumber DMS dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.

2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi API Manajemen Perubahan AMS.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws --profile saml --region us-east-1 amscm create-rfc --title "MariaDB-DMS-
Source-Endpoint" --aws-account-id ACCOUNT-ID --change-type-id ct-0attesnjqy2cx --
change-type-version 1.0 --execution-parameters "{\"Description\":\"DESCRIPTION.\",
\"VpcId\":\"VPC-ID\",\"Name\":\"MariaDB-DMS-SE\",\"Parameters\":{\"EngineName\":
\"mariadb\",\"ServerName\":\"mariadb.db.example.com\",\"Port\":3306,\"Username\":
\"DB-USER\",\"Password\":\"DB-PW\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-
pud4ghhkp7395n9bc\"}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON bernama CreateDmsSeParams .json.

```
aws amscm get-change-type-version --change-type-id "ct-0attesnjqy2cx" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsSeParams.json
```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"Description":          "MariaDB-DMS-SE",
"VpcId":                "VPC_ID",
"Name":                 "Test SE",
"StackTemplateId":      "stm-pud4ghhkp7395n9bc",
"TimeoutInMinutes":     60,
"Parameters":   {
    "Description":      "DESCRIPTION",
    "EngineName":       "mariadb",
    "ServerName":       "mariadb.db.example.com",
    "Port":             "3306",
    "Username":         "DB-USER",
    "Password":         "DB-PW",}
    }
}
```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsSeRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeRfc.json
```

4. Ubah dan simpan CreateDmsSeRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0attesnjqy2cx",
"Title":                "MariaDB-DMS-Source-Endpoint"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: CreateDmsSeRfc

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeRfc.json --execution-
parameters file://CreateDmsSeParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Sebelum Anda membuat titik akhir DMS, pastikan kata sandi Anda tidak mengandung karakter yang tidak didukung. Untuk informasi selengkapnya, lihat Membuat titik akhir sumber dan target di Panduan AWS Database Migration Service Pengguna.

Untuk mempelajari lebih lanjut, lihat Sumber untuk Migrasi Data.

Untuk titik akhir sumber S3, lihat. Titik akhir sumber DMS untuk S3: membuat

Untuk titik akhir sumber Mongo DB, lihat. Titik akhir sumber DMS untuk MongoDB: Membuat

Titik akhir sumber DMS untuk MongoDB: Membuat

Membuat DMS Mongo DB Source Endpoint dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.

3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat DMS Mongo DB Source Endpoint dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke

bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi API Manajemen Perubahan AMS.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm  --profile saml --region us-east-1 create-rfc --change-type-id
 "ct-2hxcllf1b4ey0" --change-type-version "1.0" --title 'DMS_Source_MongoDB'
 --description "DESCRIPTION"  --execution-parameters "{\"Description\":
\"DMS_MongoDB_Source_Endpoint\",\"VpcId\":\"VPC_ID\",\"Name\":\"DMS-Mongo-SE\",
\"StackTemplateId\":\"stm-pud4ghhkp7395n9bc\",\"TimeoutInMinutes\":60,\"Parameters\":
{\"DatabaseName\":\"mytestdb\",\"EngineName\":\"mongodb\",\"Port\":27017,\"ServerName
\":\"test.example.com\"}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON bernama CreateDmsSeMongoParams .json.

   ```
   aws amscm get-change-type-version --change-type-id "ct-2hxcllf1b4ey0"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateDmsSeMongoParams.json
   ```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   "Description":          "MongoDB-DMS-SE",
   "VpcId":                "VPC_ID",
   "StackTemplateId":      "stm-pud4ghhkp7395n9bc",
   "Name":                 "Test Mongo SE",
   "TimeoutInMinutes":     60,
   "Parameters":   {
       "Description":      "DESCRIPTION",
       "DatabaseName":       "mytestdb",
       "EngineName":       "mongodb",
       "ServerName":       "test.example.com",
   ```

```
    "Port":             "27017"
    }
}
```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsSeMongoRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeMongoRfc.json
```

4. Ubah dan simpan CreateDmsSeMongoRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2hxcllf1b4ey0",
"Title":                "DMS_Source_MongoDB"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: CreateDmsSeMongoRfc

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeMongoRfc.json --execution-
parameters file://CreateDmsSeMongoParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

> **Note**
>
> Anda dapat menambahkan hingga 50 tag, tetapi untuk melakukannya Anda harus mengaktifkan tampilan konfigurasi tambahan.

AMS DMS dapat menggunakan Mongo atau Relational Database Service (RDS) sebagai titik akhir sumber. Untuk titik akhir sumber S3, lihat. Titik akhir sumber DMS untuk S3: membuat

Titik akhir sumber DMS untuk S3: membuat

Membuat Endpoint Sumber DMS S3 dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat Endpoint Sumber DMS S3 dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi API Manajemen Perubahan AMS.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws --profile saml --region us-east-1 amscm create-rfc --title "S3DMSSourceEndpoint" --
aws-account-id ACCOUNT-ID --change-type-id ct-2oxl37nphsrjz --change-type-version 1.0
 --execution-parameters "{\"Description\":\"TestS3DMS-SE\",\"VpcId\":\"VPC-ID\",\"Name
\":\"S3-DMS-SE\",\"Parameters\":{\"EngineName\":\"s3\",\"S3BucketName\":\"amzn-s3-
demo-bucket\",\"S3ExternalTableDefinition\":\"{\\\"TableCount\\\":\\\"1\\\",\\\"Tables
```

```
\\\":[{\\\"TableName\\\":\\\"employee\\\",\\\"TablePath\\\":\\\"hr/employee/\\\",\\
\"TableOwner\\\":\\\"hr\\\",\\\"TableColumns\\\":[{\\\"ColumnName\\\":\\\"Id\\\",\\
\"ColumnType\\\":\\\"INT8\\\",\\\"ColumnNullable\\\":\\\"false\\\",\\\"ColumnIsPk\\\":
\\\"true\\\"},{\\\"ColumnName\\\":\\\"LastName\\\",\\\"ColumnType\\\":\\\"STRING\\\",
\\\"ColumnLength\\\":\\\"20\\\"},{\\\"ColumnName\\\":\\\"FirstName\\\",\\\"ColumnType
\\\":\\\"STRING\\\",\\\"ColumnLength\\\":\\\"30\\\"},{\\\"ColumnName\\\":\\\"HireDate\
\\\",\\\"ColumnType\\\":\\\"DATETIME\\\"},{\\\"ColumnName\\\":\\\"OfficeLocation\\\",\\
\"ColumnType\\\":\\\"STRING\\\",\\\"ColumnLength\\\":\\\"20\\\"}],\\\"TableColumnsTotal
\\\":\\\"5\\\"}]}\",\"S3ServiceAccessRoleArn\":\"arn:aws:iam::123456789101:role/ams-
ops-ct-authors-dms-s3-test-role\"},\"TimeoutInMinutes\":60,\"StackTemplateId\":\"stm-
pud4ghhkp7395n9bc\"}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON bernama CreateDmsSe S3Params.json.

   ```
   aws amscm get-change-type-version --change-type-id "ct-2oxl37nphsrjz" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsSeS3Params.json
   ```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   "Description":          "TestS3DMS-SE",
   "VpcId":                "VPC_ID",
   "Name":                 "S3-DMS-SE",
   "StackTemplateId":      "stm-pud4ghhkp7395n9bc",
   "TimeoutInMinutes":     60,
   "Parameters":   {
       "EngineName":               "s3",
       "S3BucketName":              "amzn-s3-demo-bucket",
       "S3ExternalTableDefinition":  "BUCKET-NAME",
       {"TableCount":                "1",
         "Tables":[{"TableName":"employee","TablePath":"hr/
   employee/","TableOwner":"hr","TableColumns":
   [{"ColumnName":"Id","ColumnType":"INT8","ColumnNullable":"false","ColumnIsPk":"true"},
   {"ColumnName":"LastName","ColumnType":"STRING","ColumnLength":"20"},
   {"ColumnName":"FirstName","ColumnType":"STRING","ColumnLength":"30"},
   {"ColumnName":"HireDate","ColumnType":"DATETIME"},
   {"ColumnName":"OfficeLocation","ColumnType":"STRING","ColumnLength":"20"}],"TableColumnsTot
       "S3ServiceAccessRoleArn":      "arn:aws:iam::123456789101:role/ams-ops-ct-
   authors-dms-s3-test-role",
   ```

```
      }
}
```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsSe S3rfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsSeS3Rfc.json
```

4. Ubah dan simpan file CreateDmsSe S3RFC.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2oxl37nphsrjz",
"Title":                "DMS_Source_S3"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file CreateDmsSe S3Rfc:

```
aws amscm create-rfc --cli-input-json file://CreateDmsSeS3Rfc.json --execution-
parameters file://CreateDmsSeS3Params.json
```

   Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

> **Note**
>
> Anda dapat menambahkan hingga 50 tag, tetapi untuk melakukannya Anda harus mengaktifkan tampilan konfigurasi tambahan.

AMS DMS dapat menggunakan S3 atau titik akhir sumber Relational Database Service (RDS). Untuk titik akhir sumber Mongo DB, lihat. Titik akhir sumber DMS untuk MongoDB: Membuat

4: titik akhir AWS DMS target: Buat, buat untuk S3

Anda dapat menggunakan konsol AMS atau API/CLI untuk membuat titik akhir target AMS DMS untuk berbagai database, kami menyediakan dua contoh.

Titik akhir target DMS: membuat

AMS DMS dapat menggunakan S3 atau Relational Database Service (RDS) dengan MySQL, MariaDB, Oracle, Postgresql, atau Microsoft SQL sebagai titik akhir target.

Membuat Endpoint Target DMS dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi yang lebih lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.

4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat Titik Akhir Target DMS dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-3gf8dolbo8x9p" --change-type-version "1.0" --title "TestDMSTargetEndpoint" --
execution-parameters "{\"Description\":\"TestTE\",\"VpcId\":\"VPC-ID\",\"Name\":
\"TE-NAME\",\"StackTemplateId\":\"stm-knghtmmgefafdq89u\",\"TimeoutInMinutes\":60,
\"Parameters\":{\"EngineName\":\"mysql\",\"Password\":\"testpw123\",\"Port\":\"3306\",
\"ServerName\":\"mytestdb.d5fga0rf2wpi.ap-southeast-2.rds.amazonaws.com\",\"Username\":
\"USERNAME\"}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON bernama CreateDmsTeParams .json.

   ```
   aws amscm get-change-type-version --change-type-id "ct-3gf8dolbo8x9p" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsTeParams.json
   ```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
   "Description":          "TestTE",
   "VpcId":                "VPC_ID",
   "StackTemplateId":      "stm-knghtmmgefafdq89u",
   "Name":                 "TE-NAME",
   "TimeoutInMinutes":     60,
   "Parameters":   {
       "EngineName":       "mysql",
       "ServerName":       "sql.db.example.com",
       "Port":             "3306",
       "Username":         "DB-USER",
       "Password":         "DB-PW",}
       }
   }
   ```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsTeRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsTeRfc.json
```

4. Ubah dan simpan CreateDmsTeRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-3gf8dolbo8x9p",
"Title":                "DB-DMS-Target-Endpoint"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: CreateDmsTeRfc

```
aws amscm create-rfc --cli-input-json file://CreateDmsTeRfc.json --execution-
parameters file://CreateDmsTeParams.json
```

   Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

- Jenis perubahan ini sekarang di versi 2.0.
- AMS DMS dapat menggunakan S3 atau Relational Database Service (RDS) dengan MySQL, MariaDB, Oracle, Postgresql, atau Microsoft SQL sebagai titik akhir target. Untuk titik akhir target S3, lihat. Titik akhir target DMS untuk S3: membuat
- Untuk informasi selengkapnya, lihat Target untuk Migrasi Data.
- Anda dapat menambahkan hingga 50 tag, tetapi untuk melakukannya Anda harus mengaktifkan tampilan konfigurasi tambahan.

Titik akhir target DMS untuk S3: membuat

Membuat Titik Akhir Target DMS S3 dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi yang lebih lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat Titik Akhir Target DMS S3 dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode dijelaskan di sini.

2. Kirim `aws amscm submit-rfc --rfc-id` *ID* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *ID* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-05muqzievnxk5" --change-type-version "1.0" --title "TestDMSTargetEndpointS3"
 --execution-parameters "{\"Description\":\"TestS3TE\",\"VpcId\":\"VPC-ID\",\"Name
\":\"S3TE-NAME\",\"StackTemplateId\":\"stm-knghtmmgefafdq89u\",\"TimeoutInMinutes
\":60,\"Parameters\":{\"EngineName\":\"s3\",\"S3BucketName\":\"amzn-s3-demo-bucket\",
\"S3ServiceAccessRoleArn\":\"arn:aws:iam::123456789123:role/my-s3-role\"}}"
```

TEMPLATE MEMBUAT:

1. Keluarkan parameter eksekusi untuk jenis perubahan ini ke file JSON; contoh ini menamainya CreateDmsTe S3Params.json:

```
aws amscm get-change-type-version --change-type-id "ct-05muqzievnxk5" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsTeS3Params.json
```

2. Ubah dan simpan parameter eksekusi file CreateDmsTe S3Params.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"Description":          "TestS3DMS-TE",
"VpcId":                "VPC_ID",
"StackTemplateId":      "stm-knghtmmgefafdq89u",
"Name":                 "DMS-S3-TE",
"TimeoutInMinutes":     60,
"Parameters":   {
    "EngineName":       "s3",
    "S3BucketName":      "amzn-s3-demo-bucket",
    "S3ServiceAccessRoleArn":       "arn:aws:iam::123456789101:role/ams-ops-ct-
authors-dms-s3-test-role"
     }
}
```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsTe S3rfc.json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsTeS3Rfc.json
```

4. Ubah dan simpan file CreateDmsTe S3RFC.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-05muqzievnxk5",
"Title":                "DMS_Target_S3"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file CreateDmsTe S3Rfc:

```
aws amscm create-rfc --cli-input-json file://CreateDmsTeS3Rfc.json --execution-
parameters file://CreateDmsTeS3Params.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

> **Note**
>
> Anda dapat menambahkan hingga 50 tag, tetapi untuk melakukannya Anda harus mengaktifkan tampilan konfigurasi tambahan.

AMS menyediakan jenis perubahan terpisah untuk membuat titik akhir target untuk S3. Untuk informasi selengkapnya, lihat Menggunakan Amazon S3 sebagai Target untuk AWS Database Migration Service dan Atribut Koneksi Ekstra Saat Menggunakan Amazon S3 sebagai Target untuk AWS DMS.

5: tugas AWS DMS replikasi: Buat

Anda dapat menggunakan konsol AMS atau API/CLI untuk membuat tugas AWS DMS replikasi AMS.

Buat AWS DMS tugas replikasi

Membuat Tugas AWS DMS Replikasi dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) di tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi yang lebih lama muncul di sebelah tombol Buat RFC.

- Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.

3. Pada halaman Jalankan RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.

4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, RFC berhasil membuat halaman ditampilkan dengan rincian RFC yang dikirimkan, dan output Run awal.

5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Membuat Tugas AWS DMS Replikasi dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws --profile saml --region us-east-1 amscm create-rfc --change-type-id
 "ct-1d2fml15b9eth" --change-type-version "1.0" --title "TestDMSRepTask" --
execution-parameters "{\"Description\":\"TestRepTask\",\"VpcId\":\"VPC-ID\",\"Name
\":\"DMSRepTask\",\"Parameters\":{\"CdcStartTime\":\1533776569\"MigrationType\":
\"full-load\",\"ReplicationInstanceArn\":\"REP_INSTANCE_ARN\",\"SourceEndpointArn
\":\"SOURCE_ENDPOINT_ARN\",\"TableMappings\":\"{\\\"rules\\\": [{\\\"rule-type
\\\": \\\"selection\\\",\\\"rule-id\\\": \\\"1\\\",\\\"rule-name\\\": \\\"1\\
\",\\\"object-locator\\\": {\\\"schema-name\\\": \\\"Test\\\",\\\"table-name\\
\": \\\"%\\\"}, \\\"rule-action\\\": \\\"include\\\"}] }\",\"TargetEndpointArn
\":\"TARGET_ENDPOINT_ARN\"},\"StackTemplateId\":\"stm-eos7uq0usnmeggdet\",
\"TimeoutInMinutes\":60}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON; contoh ini menamainya CreateDmsRtParams .json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-1d2fml15b9eth" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateDmsRtParams.json
   ```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"Description":          "DMSTestRepTask",
"VpcId":                "VPC_ID",
"StackTemplateId":      "stm-eos7uq0usnmeggdet",
"Name":                 "Test DMS RT",
"TimeoutInMinutes":     60,
"Parameters":   {
    "CdcStartTime":           "1533776569",
    "MigrationType":          "full-load",
    "ReplicationInstanceArn": "REP_INSTANCE_ARN",
    "SourceEndpointArn":      "SOURCE_ENDPOINT_ARN",
    "TargetEndpointArn":      "TARGET_ENDPOINT_ARN"
    "TableMappings":          {"rules": [{"rule-type": "selection","rule-id":
 "1","rule-name": "1","object-locator": {"schema-name": "Test","table-name": "%"},
 "rule-action": "include"}] }",
    }
}
```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya CreateDmsRtRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateDmsRtRfc.json
```

4. Ubah dan simpan CreateDmsRtRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1d2fml15b9eth",
"Title":                "DMS-RI-Create-RFC"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: CreateDmsRtRfc

```
aws amscm create-rfc --cli-input-json file://CreateDmsRtRfc.json --execution-
parameters file://CreateDmsRtParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Anda dapat membuat AWS DMS tugas yang menangkap tiga jenis perubahan atau data yang berbeda. Untuk informasi selengkapnya, lihat Bekerja dengan Tugas AWS DMS, Membuat Tugas, dan Membuat Tugas untuk Replikasi Berkelanjutan Menggunakan AWS DMS.

AWS DMS manajemen

AWS DMS contoh manajemen.

Mulai AWS DMS tugas replikasi

Memulai tugas AWS DMS replikasi dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.

3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Memulai tugas AWS DMS replikasi dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.
2. Kirim `aws amscm submit-rfc --rfc-id` *`ID`* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *`ID`* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke

bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc --change-type-id "ct-1yq7hhqse71yg" --change-type-version
 "1.0" --title "Start DMS Replication Task" --execution-parameters "{\"DocumentName
\":\"AWSManagedServices-StartDmsTask\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ReplicationTaskArn\":[\"TASK_ARN\"],\"StartReplicationTaskType\":[\"start-
replication\"],\"CdcStartPosition\":[\"\"],\"CdcStopPosition\":[\"\"]}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON; contoh ini menamainya StartDmsRtParams .json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-1yq7hhqse71yg" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > StartDmsRtParams.json
   ```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

   ```
   {
     "DocumentName": "AWSManagedServices-StartDmsTask",
     "Region": "us-east-1",
     "Parameters": {
         "ReplicationTaskArn": [
           "TASK_ARN"
         ],
         "StartReplicationTaskType": [
           "start-replication"
         ],
         "CdcStartPosition": [
           ""
         ],
         "CdcStopPosition": [
   ```

```
        ""
      ]
  }
}
```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya StartDmsRtRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > StartDmsRtRfc.json
```

4. Ubah dan simpan StartDmsRtRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
  "ChangeTypeId": "ct-1yq7hhqse71yg",
  "ChangeTypeVersion": "1.0",
  "Title": "Start DMS Replication Task"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: StartDmsRtRfc

```
aws amscm create-rfc --cli-input-json file://StartDmsRtRfc.json --execution-
parameters file://StartDmsRtParams.json
```

Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Anda dapat memulai tugas AWS DMS replikasi, menggunakan konsol AMS atau AMS API/CLI. Untuk informasi selengkapnya, lihat Bekerja dengan AWS DMS Tasks.

Hentikan AWS DMS tugas replikasi

Menghentikan tugas AWS DMS replikasi dengan Konsol

Tangkapan layar dari jenis perubahan ini di konsol AMS:

Cara kerjanya:

1. Arahkan ke halaman Buat RFC: Di panel navigasi kiri konsol AMS klik RFCsuntuk membuka halaman RFCs daftar, lalu klik Buat RFC.
2. Pilih jenis perubahan populer (CT) dalam tampilan default Jelajahi jenis perubahan, atau pilih CT dalam tampilan Pilih menurut kategori.
   - Jelajahi berdasarkan jenis perubahan: Anda dapat mengklik CT populer di area Buat cepat untuk segera membuka halaman Jalankan RFC. Perhatikan bahwa Anda tidak dapat memilih versi CT yang lebih lama dengan pembuatan cepat.

     Untuk mengurutkan CTs, gunakan area Semua jenis perubahan dalam tampilan Kartu atau Tabel. Di kedua tampilan, pilih CT dan kemudian klik Buat RFC untuk membuka halaman Jalankan RFC. Jika berlaku, opsi Buat dengan versi lama muncul di sebelah tombol Buat RFC.
   - Pilih berdasarkan kategori: Pilih kategori, subkategori, item, dan operasi dan kotak detail CT terbuka dengan opsi untuk Membuat dengan versi yang lebih lama jika berlaku. Klik Buat RFC untuk membuka halaman Jalankan RFC.
3. Pada halaman Run RFC, buka area nama CT untuk melihat kotak detail CT. Subjek diperlukan (ini diisi untuk Anda jika Anda memilih CT Anda di tampilan jenis perubahan Jelajahi). Buka area konfigurasi tambahan untuk menambahkan informasi tentang RFC.

   Di area konfigurasi Eksekusi, gunakan daftar drop-down yang tersedia atau masukkan nilai untuk parameter yang diperlukan. Untuk mengkonfigurasi parameter eksekusi opsional, buka area konfigurasi tambahan.
4. Setelah selesai, klik Jalankan. Jika tidak ada kesalahan, halaman RFC berhasil dibuat ditampilkan dengan detail RFC yang dikirimkan, dan output Run awal.
5. Buka area parameter Jalankan untuk melihat konfigurasi yang Anda kirimkan. Segarkan halaman untuk memperbarui status eksekusi RFC. Secara opsional, batalkan RFC atau buat salinannya dengan opsi di bagian atas halaman.

Menghentikan tugas AWS DMS replikasi dengan CLI

Cara kerjanya:

1. Gunakan Inline Create (Anda mengeluarkan `create-rfc` perintah dengan semua RFC dan parameter eksekusi disertakan), atau Template Create (Anda membuat dua file JSON, satu untuk parameter RFC dan satu untuk parameter eksekusi) dan mengeluarkan `create-rfc` perintah dengan dua file sebagai input. Kedua metode tersebut dijelaskan di sini.

2. Kirim `aws amscm submit-rfc --rfc-id` *ID* perintah RFC: dengan ID RFC yang dikembalikan.

   Pantau `aws amscm get-rfc --rfc-id` *ID* perintah RFC:.

Untuk memeriksa versi jenis perubahan, gunakan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CT_ID
```

> **Note**
>
> Anda dapat menggunakan `CreateRfc` parameter apa pun dengan RFC apa pun apakah itu bagian dari skema untuk jenis perubahan atau tidak. Misalnya, untuk mendapatkan pemberitahuan ketika status RFC berubah, tambahkan baris ini, `--notification "{\"Email\": {\"EmailRecipients\" : [\"email@example.com\"]}}"` ke bagian parameter RFC dari permintaan (bukan parameter eksekusi). Untuk daftar semua CreateRfc parameter, lihat Referensi AMS Change Management API.

BUAT SEBARIS:

Keluarkan perintah create RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris), lalu kirimkan ID RFC yang dikembalikan. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
aws amscm create-rfc --change-type-id "ct-1vd3y4ygbqmfk" --change-type-version
 "1.0" --title "Stop DMS Replication Task" --execution-parameters "{\"DocumentName
\":\"AWSManagedServices-StopDmsTask\",\"Region\":\"us-east-1\",\"Parameters\":
{\"ReplicationTaskArn\":[\"TASK_ARN\"]}}"
```

TEMPLATE MEMBUAT:

1. Output parameter eksekusi untuk jenis perubahan ini ke file JSON; contoh ini menamainya StopDmsRtParams .json:

   ```
   aws amscm get-change-type-version --change-type-id "ct-1vd3y4ygbqmfk" --query
    "ChangeTypeVersion.ExecutionInputSchema" --output text > StopDmsRtParams.json
   ```

2. Memodifikasi dan menyimpan parameter eksekusi file JSON. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
  "DocumentName": "AWSManagedServices-StopDmsTask",
  "Region": "us-east-1",
  "Parameters": {
      "ReplicationTaskArn": [
        "TASK_ARN"
      ]
  }
}
```

3. Keluarkan template JSON ke file di folder Anda saat ini; contoh ini menamainya StopDmsRtRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > StopDmsRtRfc.json
```

4. Ubah dan simpan StopDmsRtRfc file.json. Misalnya, Anda dapat mengganti konten dengan sesuatu seperti ini:

```
{
  "ChangeTypeId": "ct-1vd3y4ygbqmfk",
  "ChangeTypeVersion": "1.0",
  "Title": "Stop DMS Replication Task"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file: StopDmsRtRfc

```
aws amscm create-rfc --cli-input-json file://StopDmsRtRfc.json --execution-
parameters file://StopDmsRtParams.json
```

   Anda menerima ID RFC baru dalam respons dan dapat menggunakannya untuk mengirimkan dan memantau RFC. Sampai Anda mengirimkannya, RFC tetap dalam kondisi pengeditan dan tidak dimulai.

Kiat

Anda dapat menghentikan tugas replikasi DMS, menggunakan konsol AMS atau AMS API/CLI. Untuk informasi selengkapnya, lihat Bekerja dengan AWS DMS Tasks.

Database (DB) impor ke AMS RDS untuk Microsoft SQL Server

> **Note**
>
> Titik akhir AMS API/CLI (amscm dan amsskms) berada di Wilayah AWS N. Virginia,. `us-east-1` Bergantung pada cara autentikasi Anda disetel, dan di Wilayah AWS akun dan sumber daya Anda, Anda mungkin perlu menambahkan `--region us-east-1` saat mengeluarkan perintah. Anda mungkin juga perlu menambahkan`--profile saml`, jika itu adalah metode otentikasi Anda.

Impor DB ke AMS RDS untuk SQL Server, proses bergantung pada jenis perubahan AMS (CTs) yang dikirimkan sebagai permintaan untuk perubahan (RFCs), dan menggunakan parameter Amazon RDS API sebagai input. MicroSoft SQL Server adalah sistem manajemen basis data relasional (RDBMS). Untuk mempelajari lebih lanjut, lihat juga: Amazon Relational Database Service (Amazon RDS) dan referensi rds atau Amazon RDS API.

> **Note**
>
> Pastikan setiap RFC selesai dengan sukses sebelum melanjutkan ke langkah berikutnya.

Langkah-langkah impor tingkat tinggi:

1. Cadangkan database MS SQL lokal sumber Anda ke file.bak (cadangan)
2. Salin file.bak ke bucket transit (terenkripsi) Amazon Simple Storage Service (S3)
3. Impor .bak ke DB baru pada instans Amazon RDS MS SQL target Anda

Persyaratan:

- Tumpukan MS SQL RDS di AMS
- Tumpukan RDS dengan opsi pemulihan () `SQLSERVER_BACKUP_RESTORE`
- Ember Transit S3
- Peran IAM dengan akses bucket yang memungkinkan Amazon RDS untuk mengambil peran
- Sebuah EC2 instance dengan MS SQL Management Studio diinstal untuk mengelola RDS (dapat berupa workstation lokal)

Pengaturan

Selesaikan tugas-tugas ini untuk memulai proses impor.

1. Kirim RFC untuk membuat tumpukan RDS menggunakan Deployment | Komponen tumpukan lanjutan | Tumpukan database RDS | Buat (ct-2z60dyvto9g6c). Jangan gunakan nama DB target (`RDSDBName`parameter) dalam permintaan pembuatan, target DB akan dibuat selama impor. Pastikan untuk memberikan ruang yang cukup (`RDSAllocatedStorage`parameter). Untuk detail tentang melakukan ini, lihat Panduan Manajemen Perubahan AMS RDS DB Stack | Create.
2. Kirim RFC untuk membuat bucket transit S3 (jika belum ada) menggunakan Deployment | Komponen tumpukan lanjutan | Penyimpanan S3 | Buat (ct-1a68ck03fn98r). Untuk detail tentang melakukan ini, lihat Panduan Manajemen Perubahan AMS S3 Storage | Create.
3. Kirim Manajemen | Lainnya | Lainnya | Pembaruan (ct-1e1xtak34nx76) RFC untuk mengimplementasikan dengan rincian berikut: `customer_rds_s3_role`

   Di konsol:

   - Subjek: “Untuk mendukung MS SQL Server Database Impor, menerapkan `customer_rds_s3_role` pada account ini.
   - Nama ember Transit S3:*`BUCKET_NAME`*.
   - Informasi kontak:*`EMAIL`*.

   Dengan ImportDbParams file.json untuk CLI:

   ```
   {
          "Comment": "{"Transit S3 bucket name":"BUCKET_NAME"}",
          "Priority": "High"
        }
   }
   ```

4. Kirim Manajemen | Lainnya | Lainnya | Perbarui RFC yang meminta AMS untuk menyetel `SQLSERVER_BACKUP_RESTORE` opsi ke RDS yang dibuat pada langkah 1 (gunakan ID tumpukan dari output langkah 1, dan peran `customer_rds_s3_role` IAM dalam permintaan ini, dalam permintaan ini).
5. Kirim RFC untuk membuat EC2 instance (Anda dapat menggunakan workstation/instance yang ada EC2 atau on-premise), dan instal Microsoft SQL Management Studio pada instance tersebut.

Mengimpor database

Untuk mengimpor database (DB), ikuti langkah-langkah ini.

1. Cadangkan basis data lokal sumber Anda menggunakan pencadangan dan pemulihan MS SQL Native (lihat Support untuk pencadangan dan pemulihan asli di SQL Server). Sebagai hasil dari menjalankan operasi itu, Anda harus memiliki file.bak (cadangan).
2. Unggah file.bak ke dan bucket transit S3 yang ada menggunakan konsol AWS S3 CLI atau AWS S3. Untuk informasi tentang bucket transit S3, lihat Melindungi data menggunakan enkripsi.
3. Impor file.bak ke DB baru pada RDS target Anda untuk instans SQL Server MS SQL (untuk detail tentang jenis, lihat Amazon RDS untuk jenis instans MySQL):

    a. Masuk ke EC2 instance (stasiun kerja lokal) dan buka MS SQL Management Studio

    b. Connect ke instance RDS target yang dibuat sebagai prasyarat pada langkah #1. Ikuti prosedur ini untuk menghubungkan: Menghubungkan ke Instans DB Menjalankan Microsoft SQL Server Database Engine

    c. Mulai pekerjaan impor (pulihkan) dengan kueri Structured Query Language (SQL) baru (untuk detail tentang kueri SQL, lihat Pengantar SQL). Nama database target harus baru (jangan gunakan nama yang sama dengan database yang Anda buat sebelumnya). Contoh tanpa enkripsi:

    ```
    exec msdb.dbo.rds_restore_database
            @restore_db_name=TARGET_DB_NAME,

            @s3_arn_to_restore_from='arn:aws:s3:::BUCKET_NAME/FILENAME.bak';
    ```

    d. Periksa status pekerjaan impor secara berkala dengan menjalankan kueri ini di jendela terpisah:

    ```
    exec msdb.dbo.rds_task_status;
    ```

    Jika status berubah menjadi Gagal, cari detail kegagalan dalam pesan.

Pembersihan

Setelah Anda mengimpor database, Anda mungkin ingin menghapus sumber daya yang tidak perlu, ikuti langkah-langkah ini.

1. Hapus file cadangan (.bak) dari bucket S3. Anda dapat menggunakan konsol S3 untuk melakukan ini. Agar perintah CLI menghapus objek dari bucket S3, lihat rm di AWS CLI Command Reference.
2. Hapus bucket S3 jika Anda tidak berencana menggunakannya. Untuk langkah-langkah melakukan itu, lihat Menghapus Tumpukan.
3. Jika Anda tidak berencana untuk melakukan impor MS SQL, kirimkan Manajemen | Lainnya | Lainnya | Perbarui (ct-0xdawir96cy7k) RFC dan minta AMS menghapus peran IAM. `customer_rds_s3_role`

Penyebaran Aplikasi Tier dan Tie di AMS

Penerapan Tier dan Tie adalah tempat Anda membuat, mengonfigurasi, dan menyebarkan sumber daya tumpukan secara independen menggunakan terpisah RFCs, dan menggunakan komponen tumpukan saat Anda maju untuk mengaitkannya satu sama lain. IDs

Misalnya, untuk menerapkan situs web ketersediaan tinggi (redundan) di belakang penyeimbang beban, dan database, menggunakan pendekatan Tier dan Tie, kirimkan database, dan penyeimbang beban, dan dua instance EC2 atau grup Auto Scaling, dan konfigurasikan instance EC2 atau grup Auto Scaling dengan ID ELB yang Anda RFCs buat.

Setelah sumber daya digunakan, Anda dapat mengirimkan perubahan buat grup keamanan untuk memungkinkan sumber daya berbicara dengan database. Untuk detail tentang membuat grup keamanan, lihat Membuat Grup Keamanan.

Penerapan aplikasi tumpukan penuh di AMS

Penerapan Full Stack adalah tempat Anda mengirimkan RFC dengan CT yang membuat dan mengonfigurasi semua yang Anda butuhkan sekaligus. Misalnya, untuk menyebarkan situs web ketersediaan tinggi yang baru saja dijelaskan (EC2 instance, penyeimbang beban, dan database), Anda akan menggunakan CT yang, bersama-sama, membuat dan mengonfigurasi grup Auto Scaling, penyeimbang beban, database, dan pengaturan grup keamanan yang diperlukan agar semua instance berfungsi sebagai tumpukan. Contoh dua AMS CTs yang melakukan ini dijelaskan selanjutnya.

- High Availability Two-Tier Stack (ct-06mjngx5flwto): Jenis perubahan ini memungkinkan Anda membuat tumpukan dan mengonfigurasi Grup Auto Scaling, database yang didukung RDS, Load

Balancer, serta aplikasi dan konfigurasi. CodeDeploy Perhatikan bahwa penyeimbang beban tidak dianggap sebagai tingkatan karena dibagi di beberapa aplikasi sebagai alat jaringan dan CodeDeploy fungsinya juga dianggap sebagai alat. Selain itu, ia membuat grup CodeDeploy penyebaran (dengan nama yang Anda berikan CodeDeploy aplikasi) yang dapat digunakan untuk menyebarkan aplikasi Anda. Pengaturan grup keamanan untuk memungkinkan sumber daya berfungsi bersama dibuat secara otomatis.

- High Availability One-Tier Stack (ct-09t6q7j9v5hrn): Jenis perubahan ini memungkinkan Anda membuat tumpukan dan mengonfigurasi Grup Auto Scaling, dan Application Load Balancer. Pengaturan grup keamanan yang memungkinkan sumber daya berfungsi bersama dibuat secara otomatis.

Bekerja dengan penyediaan tipe perubahan () CTs

AMS bertanggung jawab atas infrastruktur terkelola Anda, untuk membuat perubahan, Anda harus mengirimkan RFC dengan klasifikasi CT yang benar (kategori, subkategori, item, dan operasi). Bagian ini menjelaskan cara menemukan CTs, menentukan apakah ada yang tepat untuk kebutuhan Anda, dan meminta CT baru jika tidak ada.

Lihat apakah CT yang ada memenuhi kebutuhan Anda

Setelah Anda menentukan apa yang ingin Anda terapkan dengan AMS, langkah selanjutnya adalah mempelajari yang ada CTs dan CloudFormation templat untuk melihat apakah solusi sudah ada.

Saat membuat RFC, Anda harus menentukan CT. Anda dapat menggunakan Konsol Manajemen AWS atau AMS API/CLI. Contoh penggunaan keduanya dijelaskan selanjutnya.

Anda dapat menggunakan konsol atau API/CLI untuk menemukan perubahan jenis ID (CT) atau versi. Ada dua metode, baik pencarian atau memilih klasifikasi. Untuk kedua jenis pilihan, Anda dapat mengurutkan pencarian dengan memilih Paling sering digunakan, Paling baru digunakan, atau Alfabetis.

YouTube Video: Bagaimana cara membuat RFC menggunakan AWS Managed Services CLI dan di mana saya dapat menemukan Skema CT?

Di konsol AMS, pada halaman RFCs-> Buat RFC:

- Dengan Browse by change type yang dipilih (default), baik:

- Gunakan area Quick create untuk memilih dari AMS yang paling populer CTs. Klik pada label dan halaman Jalankan RFC terbuka dengan opsi Subjek diisi otomatis untuk Anda. Selesaikan opsi yang tersisa sesuai kebutuhan dan klik Jalankan untuk mengirimkan RFC.
- Atau, gulir ke bawah ke Semua jenis perubahan area dan mulai mengetik nama CT di kotak opsi, Anda tidak harus memiliki nama jenis perubahan yang tepat atau lengkap. Anda juga dapat mencari CT dengan mengubah jenis ID, klasifikasi, atau mode eksekusi (otomatis atau manual) dengan memasukkan kata-kata yang relevan.

  Dengan tampilan Kartu default yang dipilih, kartu CT yang cocok muncul saat Anda mengetik, pilih kartu dan klik Buat RFC. Dengan tampilan Tabel dipilih, pilih CT yang relevan dan klik Buat RFC. Kedua metode membuka halaman Run RFC.

- Atau, dan untuk menjelajahi pilihan jenis perubahan, klik Pilih berdasarkan kategori di bagian atas halaman untuk membuka serangkaian kotak opsi drop-down.
- Pilih Kategori, Subkategori, Item, dan Operasi. Kotak informasi untuk jenis perubahan itu muncul panel muncul di bagian bawah halaman.
- Saat Anda siap, tekan Enter, dan daftar jenis perubahan yang cocok akan muncul.
- Pilih jenis perubahan dari daftar. Kotak informasi untuk jenis perubahan itu muncul di bagian bawah halaman.
- Setelah Anda memiliki jenis perubahan yang benar, pilih Buat RFC.

> **Note**
>
> AMS CLI harus diinstal agar perintah ini berfungsi. Untuk menginstal AMS API atau CLI, buka halaman Sumber Daya Pengembang konsol AMS. Untuk materi referensi tentang AMS CM API atau AMS SKMS API, lihat bagian Sumber Informasi AMS di Panduan Pengguna. Anda mungkin perlu menambahkan `--profile` opsi untuk otentikasi; misalnya,`aws amsskms` *`ams-cli-command`* `--profile SAML`. Anda mungkin juga perlu menambahkan `--region` opsi karena semua perintah AMS kehabisan us-east-1; misalnya. `aws amscm` *`ams-cli-command`* `--region=us-east-1`

> **Note**
>
> Titik akhir AMS API/CLI (amscm dan amsskms) berada di Wilayah AWS N. Virginia,. `us-east-1` Bergantung pada bagaimana autentikasi Anda disetel, dan di Wilayah AWS akun

dan sumber daya Anda, Anda mungkin perlu menambahkan `--region us-east-1` saat mengeluarkan perintah. Anda mungkin juga perlu menambahkan`--profile saml`, jika itu adalah metode otentikasi Anda.

Untuk mencari jenis perubahan menggunakan AMS CM API (lihat ListChangeTypeClassificationSummaries) atau CLI:

Anda dapat menggunakan filter atau kueri untuk mencari. ListChangeTypeClassificationSummaries Operasi memiliki opsi Filter untuk`Category`,`Subcategory`,`Item`, dan`Operation`, tetapi nilainya harus sama persis dengan nilai yang ada. Untuk hasil yang lebih fleksibel saat menggunakan CLI, Anda dapat menggunakan opsi. `--query`

Ubah jenis penyaringan dengan AMS CM API/CLI

Atribut	Nilai valid	Kondisi valid/default	Catatan
ChangeTypeId	String apa pun yang mewakili a ChangeTypeId (Misalnya: ct-abc123 xyz7890)	Setara	Untuk jenis perubahan IDs, lihat Referensi Ubah Jenis. Untuk jenis perubahan IDs, lihat Menemukan Jenis Perubahan atau CSIO.
Kategori	Teks bentuk bebas apa pun	Contains	Ekspresi reguler di setiap bidang individu tidak didukung. Pencarian yang tidak peka huruf besar/kecil
Subkategori			
Item			
Operasi			

1. Berikut adalah beberapa contoh klasifikasi jenis perubahan daftar:

   Perintah berikut mencantumkan semua kategori jenis perubahan.

```
aws amscm list-change-type-categories
```

Perintah berikut mencantumkan subkategori milik kategori tertentu.

```
aws amscm list-change-type-subcategories --category CATEGORY
```

Perintah berikut mencantumkan item milik kategori dan subkategori tertentu.

```
aws amscm list-change-type-items --category CATEGORY --subcategory SUBCATEGORY
```

2. Berikut adalah beberapa contoh pencarian jenis perubahan dengan kueri CLI:

   Perintah berikut mencari ringkasan klasifikasi CT untuk yang berisi “S3" di Nama item dan membuat output dari kategori, subkategori, item, operasi, dan mengubah ID tipe dalam bentuk tabel.

```
aws amscm list-change-type-classification-summaries --query
 "ChangeTypeClassificationSummaries [?contains(Item, 'S3')].
[Category,Subcategory,Item,Operation,ChangeTypeId]" --output table
```

```
+-------------------------------------------------------------+
|              ListChangeTypeClassificationSummaries          |
+----------+-------------------------+--+------+--------------+
|Deployment|Advanced Stack Components|S3|Create|ct-1a68ck03fn98r|
+----------+-------------------------+--+------+--------------+
```

3. Anda kemudian dapat menggunakan ID tipe perubahan untuk mendapatkan skema CT dan memeriksa parameternya. Perintah berikut output skema ke file JSON bernama creates3params.schema.json.

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3Params.schema.json
```

   Untuk informasi tentang penggunaan kueri CLI, lihat Cara Memfilter Output dengan Opsi --query dan referensi bahasa kueri, Spesifikasi. JMESPath

4. Setelah Anda memiliki ID tipe perubahan, kami sarankan untuk memverifikasi versi untuk jenis perubahan untuk memastikan itu adalah versi terbaru. Gunakan perintah ini untuk menemukan versi untuk jenis perubahan tertentu:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=CHANGE_TYPE_ID
```

Untuk menemukan tipe perubahan tertentu, jalankan perintah ini: `AutomationStatus`

```
aws amscm --profile saml get-change-type-version --change-type-id CHANGE_TYPE_ID --
query "ChangeTypeVersion.{AutomationStatus:AutomationStatus.Name}"
```

Untuk menemukan tipe perubahan tertentu, jalankan perintah ini: `ExpectedExecutionDurationInMinutes`

```
aws amscm --profile saml get-change-type-version --change-type-id ct-14027q0sjyt1h
 --query "ChangeTypeVersion.{ExpectedDuration:ExpectedExecutionDurationInMinutes}"
```

Setelah Anda menemukan CT yang menurut Anda sesuai, lihat parameter eksekusi skema JSON yang terkait dengannya untuk mengetahui apakah itu membahas kasus penggunaan Anda.

Gunakan perintah ini untuk menampilkan skema CT ke file JSON yang dinamai CT; contoh ini mengeluarkan skema penyimpanan Create S3:

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
--query "ChangeTypeVersion.ExecutionInputSchema" --output text >
CreateBucketParams.json
```

Mari kita lihat lebih dekat apa yang ditawarkan skema ini.

Skema Buat Bucket S3

```
{
  "$schema": "http://json-schema.org/draft-04/sch
ema#",
"name": "Create S3 Storage
"description": "Use to create an Amazon Simple
 Storage Service stack.",
  "type": "object",
```

Skema dimulai dengan CT (“deskripsi”), yang memberi tahu Anda untuk apa skema itu. Dalam hal ini, untuk membuat tumpukan penyimpanan S3.

```
  "properties": {
    "Description": {
      "description": "The description of the
 stack.",
      "type": "string",
      "minLength": 1,
      "maxLength": 500
    },
    "VpcId": {
      "description": "ID of the VPC to create the S3
 Bucket in, in the form vpc-a1b2c3d4e5f67890e.",
      "type": "string",
      "pattern": "^vpc-[a-z0-9]{17}$"
    },
    "StackTemplateId": {
      "description": "Required value: stm-s2b72
beb000000000.",
      "type": "string",
      "enum": ["stm-s2b72beb000000000"]
    },
    "Name":{
      "description": "The name of the stack to
 create.",
      "type": "string",
      "minLength": 1,
      "maxLength": 255
    },
    "Tags": {
      "description": "Up to seven tags (key/value
 pairs) for the stack.",
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "Key": {
            "type": "string",
            "minLength": 1,
            "maxLength": 127
          },
          "Value": {
            "type": "string",
            "minLength": 1,
            "maxLength": 255
          }
```

Selanjutnya, Anda memiliki properti wajib dan opsional yang dapat Anda tentukan. Nilai properti default diberikan . Properti yang diperlukan tercantum di akhir skema.

Di StackTemplateId area tersebut, Anda melihat bahwa ada satu template tumpukan khusus untuk CT dan skema ini, dan ID-nya adalah nilai properti wajib.

Skema ini memungkinkan Anda untuk menandai tumpukan yang Anda buat, untuk tujuan pembukuan internal. Selain itu, beberapa opsi, seperti cadangan, memerlukan tag key:Backup dan Value:True. Untuk informasi mendalam, baca Menandai Sumber Daya Amazon EC2 Anda.

```
        },
        "additionalProperties": false,
        "required": [
          "Key",
          "Value"
        ]
      },
      "minItems": 1,
      "maxItems": 7
    },
    "TimeoutInMinutes": {
      "description": "The amount of time, in minutes,
 to allow for creation of the stack.",
      "type": "number",
      "minimum": 0,
      "maximum": 60
    },
    "Parameters": {
      "description": "Specifications for the
 stack.",
      "type": "object",
      "properties": {
        "AccessControl": {
          "description": "The canned (predefined)
 access control list (ACL) to assign to the bucket.",
          "type": "string",
          "enum": [
            "Private",
            "PublicRead",
            "AuthenticatedRead",
            "BucketOwnerRead"
          ]
        },
        "BucketName": {
          "description": "A name for the bucket.
 The bucket name must contain only lowercase letters,
 numbers, periods (.), and hyphens (-).",
          "type": "string",
          "pattern": "^[a-z0-9]([-.a-z0-9]+)[a-z
0-9]$",
          "minLength": 3,
          "maxLength": 63
        }
      },
```

Bagian Parameter dari skema CT JSON adalah tempat Anda memberikan parameter eksekusi.

Untuk skema ini, hanya ACL dan parameter eksekusi BucketName yang diperlukan.

```
      "additionalProperties": false,
      "required": [
        "AccessControl",
        "BucketName"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "Description",
    "VpcId",
    "StackTemplateId",
    "Name",
    "TimeoutInMinutes",
    "Parameters"
  ]
}
```

Minta CT baru

Setelah memeriksa skema, Anda dapat memutuskan bahwa itu tidak menyediakan parameter yang cukup untuk membuat penerapan yang Anda inginkan. Jika itu masalahnya, periksa CloudFormation templat yang ada untuk menemukan templat yang lebih dekat dengan yang Anda inginkan. Setelah Anda tahu parameter tambahan apa yang Anda butuhkan, kirimkan Manajemen | Lainnya | Lainnya | Buat CT.

> **Note**
>
> Semua Lainnya | Buat dan Pembaruan Lainnya CTs menerima perhatian dari operator AMS, yang akan menghubungi Anda untuk mendiskusikan CT baru.

Untuk mengirimkan permintaan CT baru, akses konsol AMS melalui reguler Konsol Manajemen AWSdan kemudian ikuti langkah-langkah ini.

1. Dari navigasi kiri, klik RFCs.

   Halaman RFCs dasbor terbuka.

2. Klik Buat.

Halaman Buat permintaan untuk perubahan terbuka.

3. Pilih Manajemen di Kategori daftar drop-down, dan Lainnya untuk Subkategori dan Item. Untuk Operasi, pilih Buat. RFC akan membutuhkan persetujuan sebelum dapat diimplementasikan.
4. Masukkan informasi mengapa Anda menginginkan CT, misalnya: Meminta CT penyimpanan Create S3 yang dimodifikasi yang memungkinkan kustom ACLs, berdasarkan CT penyimpanan Create S3 yang ada. Ini akan menghasilkan CT baru: Deployment | Komponen Tumpukan Tingkat Lanjut | Penyimpanan S3 | Buat ACL kustom S3. CT baru ini bisa bersifat publik.
5. Klik Kirim.

   RFC Anda ditampilkan di dasbor RFC.

Uji CT baru

Setelah AWS Managed Services membuat CT baru itu, Anda mengujinya dengan mengirimkan RFC dengannya. Jika Anda bekerja dengan AMS untuk membuat CT baru telah disetujui sebelumnya, maka Anda cukup mengikuti pengiriman RFC standar, dan perhatikan hasilnya (untuk detail tentang pengiriman RFCs, lihat Membuat dan Mengirimkan RFC). Jika CT baru tidak disetujui sebelumnya (Anda ingin memastikan bahwa CT tidak pernah dijalankan tanpa persetujuan eksplisit), maka Anda perlu mendiskusikan implementasinya dengan AMS setiap kali Anda ingin menjalankannya.

Mulai cepat

Topik

Menggunakan kombinasi jenis perubahan AMS, Anda dapat menyelesaikan tugas yang kompleks.

Anda dapat menggunakan sistem manajemen perubahan AMS untuk menyiapkan Penjadwal Sumber Daya AMS, untuk landing zone multi-akun (MALZ) atau untuk akun landing zone (SALZ) akun tunggal. Prosesnya bervariasi. Juga, untuk melakukan transfer file dan snapshot lintas akun.

Penjadwal Sumber Daya AMS mulai cepat

Gunakan panduan mulai cepat ini untuk menerapkan Penjadwal Sumber Daya AMS, penjadwal instans berbasis tag untuk menghemat biaya di AMS Advanced.

Penjadwal Sumber Daya AMS didasarkan pada AWS Instance Scheduler.

Terminologi Penjadwal Sumber Daya AMS

Sebelum Anda mulai, ada baiknya Anda mengetahui terminologi AMS Resource Scheduler:

- periode: Setiap jadwal harus berisi setidaknya satu periode yang menentukan waktu instance harus dijalankan. Jadwal dapat berisi lebih dari satu periode. Ketika lebih dari satu periode digunakan dalam jadwal, Resource Scheduler menerapkan tindakan awal yang sesuai ketika setidaknya salah satu aturan periode benar.
- zona waktu: Untuk daftar nilai zona waktu yang dapat diterima untuk digunakan dalam DefaultTimezoneparameter yang direferensikan nanti, lihat kolom TZ dari Daftar Zona Waktu Database TZ.
- hibernasi: Ketika disetel ke EC2 instance true yang diaktifkan untuk hibernasi dan memenuhi persyaratan hibernasi hibernasi hibernasi (). suspend-to-disk Periksa EC2 konsol untuk mengetahui apakah instance Anda diaktifkan untuk hibernasi. Gunakan hibernasi untuk EC2 instans Amazon yang dihentikan yang menjalankan Amazon Linux.
- diberlakukan: Ketika disetel ke true, berdasarkan jadwal yang ditentukan, Resource Scheduler menghentikan sumber daya yang sedang berjalan jika sumber daya dimulai secara manual di

luar periode berjalan, dan memulai sumber daya jika dihentikan secara manual selama periode berjalan.

- retain_running: Ketika disetel ke true, mencegah Resource Scheduler menghentikan instance di akhir periode berjalan jika instance dimulai secara manual sebelum awal periode. Misalnya, jika instance dengan periode terkonfigurasi yang berjalan dari jam 9 pagi sampai jam 5 sore dimulai secara manual sebelum jam 9 pagi, Resource Scheduler tidak menghentikan instance pada jam 5 sore.
- ssm-maintenance-window: Tambahkan jendela AWS Systems Manager pemeliharaan sebagai periode berjalan ke jadwal. Saat Anda menentukan nama jendela pemeliharaan yang ada di akun dan Wilayah AWS yang sama dengan tumpukan yang diterapkan untuk menjadwalkan EC2 instans Amazon Anda, Resource Scheduler akan memulai instance sebelum dimulainya jendela pemeliharaan dan menghentikan instance di akhir jendela pemeliharaan, jika tidak ada periode berjalan lainnya yang menentukan bahwa instance harus berjalan, dan jika peristiwa pemeliharaan selesai.

  Resource Scheduler menggunakan AWS Lambda frekuensi yang Anda tentukan selama konfigurasi awal untuk menentukan berapa lama sebelum jendela pemeliharaan memulai instance Anda. Jika Anda menyetel AWS CloudFormation parameter Frekuensi ke 10 menit atau kurang, Resource Scheduler memulai instance 10 menit sebelum jendela pemeliharaan. Jika Anda mengatur frekuensi menjadi lebih dari 10 menit, Resource Scheduler memulai instance dengan jumlah menit yang sama dengan frekuensi yang Anda tentukan. Misalnya, jika Anda menyetel frekuensi jendela pemeliharaan Systems Manager menjadi 30 menit, Resource Schedulers memulai instance 30 menit sebelum jendela pemeliharaan.

  Untuk informasi selengkapnya, lihat AWS Systems Manager Pemeliharaan Windows.
- override-status: Ganti sementara tindakan awal dan hentikan Jadwal Penjadwal yang dikonfigurasi oleh Resource Scheduler. Jika Anda menyetel bidang untuk berjalan, Resource Scheduler akan dimulai, tetapi tidak berhenti, instance yang berlaku. Instance berjalan sampai Anda menghentikannya secara manual. Jika Anda menyetel status penggantian menjadi berhenti, Resource Scheduler akan berhenti tetapi tidak memulai instance yang berlaku. Instance tidak berjalan sampai Anda memulainya secara manual.

Implementasi Penjadwal Sumber Daya AMS

Untuk menerapkan solusi penjadwal Sumber Daya AMS, ikuti langkah-langkah berikut.

1. Kirim Penerapan | Penjadwal Sumber Daya AMS | Solusi | Terapkan (ct-0ywnhc8e5k9z5) RFC dan berikan parameter berikut:

   - SchedulingActive: Ya untuk mengaktifkan penjadwalan sumber daya, Tidak untuk menonaktifkan. Default-nya adalah Ya.
   - ScheduledServices: Masukkan daftar layanan yang dipisahkan koma untuk menjadwalkan sumber daya. Nilai yang valid termasuk kombinasi penskalaan otomatis, ec2, dan rds. Defaultnya adalah penskalaan otomatis, ec2, rds.
   - TagName: Nama Tag Key yang mengaitkan skema jadwal sumber daya dengan sumber daya layanan. Defaultnya adalah Jadwal.

     > **Note**
     >
     > Penyebaran Resource Scheduler Anda hanya akan beroperasi pada sumber daya yang memiliki tag ini.

   - DefaultTimezone: Nama zona waktu, dalam bentuk AS/Pasifik, untuk digunakan sebagai zona waktu default. Defaultnya adalah UTC.
2. Setelah Anda menerima konfirmasi bahwa RFC pada langkah pertama berhasil dijalankan, Anda dapat mengirimkan Periode | Tambahkan jenis perubahan.
3. Terakhir, kirimkan RFC untuk menambahkan jadwal ke periode yang dibuat pada langkah kedua. Gunakan Jadwal | Tambahkan jenis perubahan.

Implementasi dan penggunaan Penjadwal Sumber Daya AMS FAQs

Pertanyaan yang sering diajukan tentang Penjadwal Sumber Daya AMS.

T: Apa yang terjadi jika saya mengaktifkan hibernasi tetapi EC2 instance tidak mendukungnya?

J: Hibernasi menyimpan konten dari memori instans (RAM) ke volume root Amazon Elastic Block Store (Amazon EBS) Elastic Block Store (Amazon EBS). Jika bidang ini disetel ke true, instance akan hibernasi saat Resource Scheduler menghentikannya.

Jika Anda menyetel Resource Scheduler untuk menggunakan hibernasi tetapi instance Anda tidak diaktifkan untuk hibernasi atau tidak memenuhi prasyarat hibernasi, Resource Scheduler mencatat peringatan dan instance dihentikan tanpa hibernasi. Untuk informasi selengkapnya, lihat Hibernasi Instance Anda.

T: Apa yang terjadi jika saya menyetel override_status dan enforced?

A: Jika Anda menyetel override_status untuk berjalan dan menyetel diberlakukan ke true (mencegah instance dimulai secara manual di luar periode berjalan), Resource Scheduler menghentikan instance.

Jika Anda menyetel override_status ke stop, dan menetapkan enforced ke true (mencegah instance dihentikan secara manual selama periode berjalan), Resource Scheduler akan memulai ulang instance.

> **Note**
>
> Jika diberlakukan adalah false, perilaku override yang dikonfigurasi akan diterapkan.

T: Setelah Penjadwal Sumber Daya AMS diterapkan, bagaimana cara menonaktifkan atau mengaktifkan penjadwal sumber daya di akun saya?

J: Untuk menonaktifkan atau mengaktifkan Penjadwal Sumber Daya AMS:

- Untuk menonaktifkan: Buat RFC menggunakan Status | Nonaktifkan. Pastikan untuk mengatur SchedulerStateke DISABLE
- Untuk mengaktifkan: Buat RFC menggunakan Status | Aktifkan. Pastikan untuk mengatur SchedulerStateke ENABLE

T Apa yang terjadi jika periode Penjadwal Sumber Daya AMS termasuk dalam jendela pemeliharaan tambalan saya?

J: Resource Scheduler bekerja berdasarkan jadwal yang dikonfigurasi. Jika dikonfigurasi untuk menghentikan instance saat penambalan sedang dalam penerbangan, maka itu menghentikan instance kecuali jendela penambalan ditambahkan sebagai periode ke jadwal sebelum penambalan dimulai. Dengan kata lain, Resource Scheduler tidak memulai otomatis instance yang dihentikan untuk ditambal kecuali periode yang ditentukan dikonfigurasi. Untuk menghindari konflik dengan jendela pemeliharaan tambalan Anda, tambahkan jendela waktu yang dialokasikan untuk menambal ke jadwal Resource Scheduler sebagai titik. Untuk menambahkan periode ke jadwal yang ada, buat RFC menggunakan Periode | Tambah.

T Jika saya perlu memiliki jadwal yang berbeda untuk EC2 instans yang berbeda, dapatkah saya memiliki lebih dari satu pengaturan jadwal di dalam akun saya?

A: Ya, Anda dapat membuat beberapa jadwal. Setiap jadwal dapat memiliki beberapa periode berdasarkan persyaratan. Ketika Penjadwal Sumber Daya AMS diaktifkan di akun, Kunci Tag dikonfigurasi. Sebagai contoh, jika Kunci Tag adalah “Jadwal”, Nilai Tag dapat berbeda berdasarkan jadwal berbeda yang sesuai dengan nama jadwal Penjadwal Sumber Daya AMS. Untuk menambahkan jadwal baru, Anda dapat membuat RFC menggunakan tipe perubahan Management | AMS Resource Scheduler | Schedule | Add (ct-2bxelbn765ive), lihat Jadwal | Tambah.

T: Di mana saya dapat menemukan semua jenis perubahan berbeda yang didukung untuk Penjadwal Sumber Daya AMS?

J: AMS memiliki jenis perubahan Resource Scheduler untuk menyebarkan Penjadwal Sumber Daya AMS ke akun Anda; mengaktifkan atau menonaktifkannya; menentukan, menambah, memperbarui, dan menghapus jadwal dan periode yang akan digunakan dengannya; dan jelaskan (dapatkan deskripsi terperinci tentang) jadwal dan periode.

Menyiapkan pencadangan lintas akun (intra-wilayah)

AWS Backup mendukung kemampuan untuk menyalin snapshot dari satu akun ke akun lain dalam Wilayah AWS yang sama selama kedua akun tersebut berada dalam Organisasi AWS yang sama. Sebagai contoh, di AMS Advanced multi-account landing zone (MALZ), Anda dapat mengatur salinan snapshot lintas akun dalam Wilayah AWS yang sama menggunakan start cepat ini.

Untuk informasi selengkapnya, lihat AWS Backup dan AWS Organizations menghadirkan fitur pencadangan lintas akun

Anda menyalin snapshot lintas akun untuk pemulihan bencana (DR). Anda mungkin memiliki persyaratan untuk menyimpan snapshot dalam Wilayah AWS yang sama, tetapi di seberang batas akun, untuk perlindungan data.

Ikhtisar:

Pada tingkat tinggi, ini adalah langkah-langkah untuk pencadangan lintas akun dalam AMS:

- Buat akun tujuan untuk meng-host backup di Wilayah AWS tempat landing zone AMS Anda di-host (langkah 1)
- Buat kunci KMS untuk mengenkripsi cadangan di akun tujuan (langkah 3)
- Buat brankas cadangan di akun tujuan di wilayah yang sama dengan landing zone AMS Advanced Anda (langkah 4)

- Aktifkan pengaturan lintas akun di akun Manajemen Anda (langkah 5)
- Membuat atau memodifikasi rencana dan aturan cadangan akun sumber (langkah 6)

> **Note**
>
> Pastikan akun sumber dan tujuan berada di Wilayah yang sama. Jika Anda ingin menyalin cadangan lintas wilayah, hubungi CA atau CSDM Anda.

Untuk mengaktifkan dan mengatur cadangan lintas akun:

1. Buat akun tujuan untuk meng-host cadangan; jika Anda sudah memiliki akun seperti itu, Anda dapat melewati langkah ini. Untuk membuat akun, kirimkan RFC dari akun Management Payer Anda menggunakan Deployment | Managed landing zone | Managed landing zone | Akun manajemen | Buat akun aplikasi (dengan VPC) jenis perubahan (ct-1zdasmc2ewzrs).
2. [Opsional] Jika sumber daya atau snapshot dienkripsi di akun sumber (misalnya, Prod), bagikan kunci KMS yang digunakan untuk enkripsi dengan akun tujuan. Untuk melakukan ini, kirimkan RFC menggunakan Manajemen | Komponen tumpukan lanjutan | Kunci KMS | Perbarui jenis perubahan (ct-3ovo7px2vsa6n).
3. Di akun tujuan, buat Kunci KMS yang akan digunakan untuk enkripsi Backup Vault. Untuk melakukan ini, kirimkan RFC menggunakan Deployment | Komponen tumpukan lanjutan | Kunci KMS | Buat (auto) jenis perubahan (ct-1d84keiri1jhg).
4. Di akun tujuan, buat Brankas Cadangan menggunakan kunci yang dibuat sebelumnya. AWS Backup Vaults dapat dibuat dengan menggunakan jenis perubahan otomatis CFN ingest, Deployment | Ingestion | Stack from CloudFormation Template | Create (ct-36cn2avfrrj9v). Dalam permintaan yang sama, kebijakan akses vault perlu dimodifikasi agar akun sumber mengakses vault. Berikut adalah contoh kebijakan:

   Contoh CloudFormation template untuk Backup Vault:

```
{
  "Description": "Test infrastructure",
  "Resources": {
  "BackupVaultForTesting": {
    "Type": "AWS::Backup::BackupVault",
    "Properties": {
      "BackupVaultName": "backup-vault-for-test",
```

```
      "EncryptionKeyArn" : "arn:aws:kms:us-east-2:123456789012:key/227d8xxx-
aefx-44ex-a09x-b90c487b4xxx",
        "AccessPolicy" : {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Sid": "AllowSrcAccountPermissionsToCopy",
              "Effect": "Allow",
              "Action": "backup:CopyIntoBackupVault",
              "Resource": "*",
              "Principal": {
                "AWS": ["arn:aws:iam::987654321098:root"]
              }
            }
          ]
        }
      }
    }
  }
}
```

5. Dari akun Management Payer Anda, aktifkan backup Cross-Account. Untuk melakukan ini, kirimkan RFC menggunakan Manajemen | AWS Backup | Rencana cadangan | Aktifkan salinan lintas akun (Akun manajemen) jenis perubahan (ct-2yja7ihh30ply).
6. Terakhir, dari akun sumber tempat cadangan bersumber, buat aturan atau aturan rencana cadangan yang mengatur cadangan untuk menyalin snapshot lintas akun. Untuk melakukannya, kirimkan RFC menggunakan Deployment | AWS Backup | Backup plan | Buat jenis perubahan (ct-2hyozbpa0sx0m). Jika Anda perlu memperbarui rencana cadangan yang ada, kirimkan RFC menggunakan Manajemen | Lainnya | Lainnya | Perbarui jenis perubahan (ct-0xdawir96cy7k) dengan informasi ini:

    1. Nama paket cadangan serta nama aturan yang akan diperbarui.
    2. Brankas cadangan destination/ICE akun ARN.
    3. Retensi yang days/months Anda inginkan untuk menyimpan snapshot di lemari besi ICE target.

Tutorial

Topik

Tutorial berikut merinci langkah-langkah untuk membuat tumpukan dua tingkat dengan Ketersediaan Tinggi (ct-06mjngx5flwto), menggunakan CLI dan menggunakan Konsol dan menerapkan grup Auto Scaling Amazon Linux atau RHEL (ASG). EC2 tier-and-tieTutorial serupa mengikuti masing-masing (satu untuk Konsol dan satu untuk CLI), yang menggunakan terpisah CTs, dibuat sedemikian rupa sehingga memungkinkan Anda untuk mengikat sumber daya saat dibuat.

Deskripsi untuk semua opsi CT, termasuk ChangeTypeId dapat ditemukan di Referensi Tipemanagedservices/latest/ctref/Ubah.

Tutorial Konsol: Ketersediaan Tinggi Dua Tingkat Stack (Linux/RHEL)

Bagian ini menjelaskan cara menerapkan WordPress situs ketersediaan tinggi (HA) ke lingkungan AMS menggunakan konsol AMS.

> **Note**
>
> Panduan penerapan ini telah diuji di lingkungan AMZN Linux dan RHEL.

Ringkasan tugas dan diperlukan RFCs:

1. Buat infrastruktur (tumpukan dua tingkat HA)
2. Buat bucket S3 untuk aplikasi CodeDeploy
3. Buat bundel WordPress aplikasi dan unggah ke bucket S3
4. Menyebarkan aplikasi dengan CodeDeploy
5. Akses WordPress situs dan masuk untuk memvalidasi penerapan

6. Meruntuhkan penyebaran

Deskripsi untuk semua opsi CT, termasuk ChangeTypeId, dapat ditemukan di Referensi Jenis Perubahan AMS.

Sebelum Anda Memulai

Deployment | Advanced Stack Components | High Availability Two Tier Stack | Create CT membuat grup Auto Scaling, load balancer, database, dan CodeDeploy nama aplikasi serta grup deployment (dengan nama yang sama dengan yang Anda berikan pada aplikasi). Untuk informasi tentang CodeDeploy lihat Apa itu CodeDeploy?

Panduan ini menggunakan RFC Stack Dua Tingkat Ketersediaan Tinggi yang menyertakan `UserData` dan juga menjelaskan cara membuat WordPress bundel yang dapat digunakan. CodeDeploy

Yang `UserData` ditunjukkan dalam contoh mendapatkan metadata instance seperti ID instance, wilayah, dll, dari dalam instance yang sedang berjalan dengan menanyakan layanan metadata EC2 instance yang tersedia di http://169.254.169.254/latest/meta-data/. Baris ini dalam skrip data pengguna:`REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`, mengambil nama zona ketersediaan dari layanan meta-data ke variabel $REGION untuk wilayah yang kami dukung, dan menggunakannya untuk melengkapi URL untuk bucket S3 tempat agen diunduh. CodeDeploy IP 169.254.169.254 hanya dapat dirutekan dalam VPC (semua dapat menanyakan layanan). VPCs Untuk informasi tentang layanan, lihat Metadata Instans dan Data Pengguna. Perhatikan juga bahwa skrip yang UserData dimasukkan sebagai dijalankan sebagai pengguna “root” dan tidak perlu menggunakan perintah “sudo”.

Panduan ini meninggalkan parameter berikut pada nilai default (ditampilkan):

- Grup Auto Scaling:. `Cooldown=300, DesiredCapacity=2, EBSOptimized=false, HealthCheckGracePeriod=600, IAMInstanceProfile=customer-mc-ec2-instance-profile, InstanceDetailedMonitoring=true, InstanceRootVolumeIops=0, InstanceRootVolumeType=standard, InstanceType=m3.medium, MaxInstances=2, MinInstances=2, ScaleDownPolicyCooldown=300, ScaleDownPolicyEvaluationPeriods=4, ScaleDownPolicyPeriod=60, ScaleDownPolicyScalingAdjustment=-1, ScaleDownPolicyStatistic=Average, ScaleDownPolicyThreshold=35, ScaleMetricName=CPUUtilization,`

`ScaleUpPolicyCooldown=60, ScaleUpPolicyEvaluationPeriods=2, ScaleUpPolicyPeriod=60, ScaleUpPolicyScalingAdjustment=2, ScaleUpPolicyStatistic=Average, ScaleUpPolicyThreshold=75`

- Load Balancer:. `HealthCheckInterval=30, HealthCheckTimeout=5`
- Basis data:`BackupRetentionPeriod=7, Backups=true, InstanceType=db.m3.medium, IOPS=0, MultiAZ=true, PreferredBackupWindow=22:00-23:00, PreferredMaintenanceWindow=wed:03:32-wed:04:02, StorageEncrypted=false, StorageEncryptionKey="", StorageType=gp2`.
- Aplikasi:`DeploymentConfigName=CodeDeployDefault.OneAtATime`.

Parameter Variabel:

Konsol menyediakan opsi ASAP untuk waktu mulai dan panduan ini merekomendasikan untuk menggunakannya. ASAP menyebabkan RFC dieksekusi segera setelah persetujuan disahkan.

> **Note**
>
> Ada banyak parameter yang mungkin Anda pilih untuk diatur secara berbeda dari seperti yang ditunjukkan. Nilai untuk parameter yang ditunjukkan dalam contoh telah diuji tetapi mungkin tidak tepat untuk Anda. Hanya nilai yang diperlukan yang ditampilkan dalam contoh. Nilai dalam *`replaceable`* font harus diubah karena mereka khusus untuk akun Anda.

Buat Infrastruktur

Prosedur ini menggunakan CT stack dua tingkat ketersediaan tinggi diikuti oleh CT penyimpanan Create S3.

Mengumpulkan data berikut sebelum Anda mulai akan membuat penyebaran berjalan lebih cepat.

DATA YANG DIBUTUHKAN HA STACK:

- AutoScalingGroup:
  - UserData: Nilai ini disediakan dalam tutorial ini. Ini termasuk perintah untuk mengatur sumber daya untuk CodeDeploy dan memulai CodeDeploy agen.
  - AMI-ID: Nilai ini menentukan sistem operasi EC2 instance grup Auto Scaling (ASG) Anda akan berputar. Pilih AMI di akun Anda yang dimulai dengan “pelanggan-” dan merupakan sistem

operasi yang Anda inginkan. Temukan AMI IDs di AMS Console VPCs -> halaman VPCs detail. Panduan ini untuk ASGs dikonfigurasi untuk menggunakan Amazon Linux atau RHEL AMI.

- Database:
  - Parameter ini DBEngine, EngineVersion,, dan LicenseModelharus diatur sesuai dengan situasi Anda meskipun nilai yang ditunjukkan dalam contoh telah diuji. Tutorial menggunakan nilai-nilai ini, masing-masing:*MySQL*,*8.0.16*,*general-public-license*.
  - Parameter ini DBName, MasterUserPassword,, dan MasterUsernamediperlukan saat menerapkan bundel aplikasi. Tutorial menggunakan nilai-nilai ini, masing-masing:*wordpressDB*,*p4ssw0rd*,*admin*. Perhatikan bahwa hanya DBName dapat berisi karakter alfanumerik.
  - Ketika Anda memasukkan MasterUsernameuntuk RDS DB, itu akan muncul di cleartext, jadi masuk ke database sesegera mungkin dan ubah kata sandi untuk memastikan keamanan Anda.
  - Untuk RDSSubnetId, gunakan dua subnet Private. Masukkan mereka satu per satu menekan “Enter” setelah masing-masing. Temukan Subnet IDs dengan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (CLI list-subnet-summaries:) atau di halaman AMS Console -> VPC details. VPCs
- LoadBalancer:
  - Atur parameter ini, Public ke true karena tutorial menggunakan subnet ELB Public.
  - ELBSubnetIds: Gunakan dua subnet Publik. Masukkan mereka satu per satu menekan “Enter” setelah masing-masing. Temukan Subnet IDs dengan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (CLI list-subnet-summaries:) atau di halaman AMS Console -> VPC details. VPCs
- Aplikasi: ApplicationNameNilai menetapkan nama CodeDeploy aplikasi dan nama grup CodeDeploy penyebaran. Anda menggunakannya untuk menyebarkan aplikasi Anda. Itu harus unik di akun. Untuk memeriksa CodeDeploy nama akun Anda, lihat CodeDeploy Konsol. Contoh menggunakan *WordPress* tetapi, jika Anda akan menggunakan nilai itu, pastikan bahwa itu belum digunakan.

1. Luncurkan tumpukan ketersediaan tinggi.

   a. Pada halaman Buat RFC, pilih kategori Deployment, subkategori Stack Standar, item Ketersediaan tinggi tumpukan dua tingkat dan operasi Buat, dari daftar.

   b. PENTING: Pilih Advanced dan atur nilai seperti yang ditunjukkan.

Anda hanya perlu memasukkan nilai untuk opsi berbintang (*), nilai yang diuji ditampilkan dalam contoh; Anda dapat membiarkan opsi kosong yang tidak diperlukan kosong.

c. Untuk bagian Deskripsi RFC:

```
Subject: WP-HA-2-Tier-RFC
```

d. Untuk bagian Informasi sumber daya, tetapkan parameter untuk AutoScalingGroup, Database LoadBalancer,, Aplikasi, dan Tag.

Juga, tujuan dari kunci tag AppName "" adalah agar Anda dapat dengan mudah mencari instance ASG di EC2 konsol; Anda dapat memanggil kunci tag ini “Nama” atau nama kunci lain yang Anda inginkan. Perhatikan bahwa Anda dapat menambahkan hingga 50 tag.

```
UserData:
    #!/bin/bash
    REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/
 | sed 's/[a-z]$//')
    yum -y install ruby httpd
    chkconfig httpd on
    service httpd start
    touch /var/www/html/status
    cd /tmp
    curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/install
    chmod +x ./install
    ./install auto
    chkconfig codedeploy-agent on
    service codedeploy-agent start
AmiId:              AMI-ID
Description:        WP-HA-2-Tier-Stack

Database:
    LicenseModel:    general-public-license (USE RADIO BUTTON)
    EngineVersion:   8.0.16
    DBEngine:        MySQL
    RDSSubnetIds:    PRIVATE_AZ1 PRIVATE_AZ2 (ENTER ONE AT A TIME PRESSING
 "ENTER" AFTER EACH)
    MasterUserPassword:  p4ssw0rd
    MasterUsername:      admin
    DBName:              wordpressDB

LoadBalancer:
```

```
    Public:              true (USE RADIO BUTTON)
    ELBSubnetIds:        PUBLIC_AZ1 PUBLIC_AZ2

Application:
    ApplicationName:     WordPress

Tags:
    Name:                WP-Rhel-Stack
```

e. Klik Kirim setelah selesai.

2. Masuk ke database yang Anda buat dan ubah kata sandi.

3. Luncurkan S3 bucket Stack.

   Mengumpulkan data berikut sebelum Anda mulai akan membuat penyebaran berjalan lebih cepat.

   EMBER DATA S3 YANG DIBUTUHKAN:

   - VPC-ID: Nilai ini menentukan di mana Bucket S3 Anda akan berada. Temukan VPC IDs dengan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (CLI:) atau di halaman AMS Console. list-vpc-summaries VPCs
   - BucketName: Nilai ini menetapkan nama Bucket S3, Anda menggunakannya untuk mengunggah bundel aplikasi Anda. Itu harus unik di seluruh wilayah akun dan tidak dapat menyertakan huruf besar. Menyertakan ID akun Anda sebagai bagian dari BucketName bukan persyaratan tetapi membuatnya lebih mudah untuk mengidentifikasi bucket nanti. Untuk melihat nama bucket S3 yang ada di akun, buka Konsol Amazon S3 untuk akun Anda.

   a. Pada halaman Create RFC, pilih kategori Deployment, subkategori Advanced Stack Components, item S3 storage, dan operation Create dari RFC CT pick list.

   b. Pertahankan opsi Basic default dan atur nilai seperti yang ditunjukkan.

```
Subject:             S3-Bucket-WP-HA-RFC
Description:         S3BucketForWordPressBundles
BucketName:          ACCOUNT_ID-BUCKET_NAME
AccessControl:       Private
VpcId:               VPC_ID
Name:                S3-Bucket-WP-HA-Stack
TimeoutInMinutes:    60
```

c. Klik Kirim setelah selesai. Bucket yang digunakan dengan jenis perubahan ini memungkinkan read/write akses penuh ke seluruh akun.

Membuat, Mengunggah, dan Menyebarkan Aplikasi

Pertama, buat bundel WordPress aplikasi, lalu gunakan CodeDeploy CTs untuk membuat dan menyebarkan aplikasi.

1. Unduh WordPress, ekstrak file dan buat file. /scripts direktori.

   Perintah Linux:

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

   Windows: Tempel `https://github.com/WordPress/WordPress/archive/master.zip` ke jendela browser dan unduh file zip.

   Buat direktori sementara untuk merakit paket.

   Linux:

   ```
   mkdir /tmp/WordPress
   ```

   Windows: Buat direktori WordPress "", Anda akan menggunakan jalur direktori nanti.

2. Ekstrak WordPress sumber ke direktori WordPress "" dan buat file. /scripts direktori.

   Linux:

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows: Buka direktori "WordPress" yang Anda buat dan buat direktori “skrip” di sana.

   Jika Anda berada di lingkungan Windows, pastikan untuk mengatur jenis istirahat untuk file skrip ke Unix (LF). Di Notepad ++, ini adalah opsi di kanan bawah jendela.

3. Buat file CodeDeploy appspec.yl, di WordPress direktori (jika menyalin contoh, periksa lekukan, setiap spasi dihitung). PENTING: Pastikan jalur “sumber” benar untuk menyalin WordPress file (dalam hal ini, di WordPress direktori Anda) ke tujuan yang diharapkan (/var/www/html/WordPress). Dalam contoh, file appspec.ymlada di direktori dengan WordPress file, jadi hanya “/” yang diperlukan. Juga, bahkan jika Anda menggunakan RHEL AMI untuk grup Auto Scaling Anda, biarkan baris “os: linux” apa adanya. Contoh file appspec.yml:

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

4. Buat skrip file bash di file. WordPress /scripts direktori.

   Pertama, buat `config_wordpress.sh` dengan konten berikut (jika Anda mau, Anda dapat mengedit file wp-config.php secara langsung).

   > **Note**
   >
   > Ganti *`DBName`* dengan nilai yang diberikan dalam HA Stack RFC (misalnya,`wordpress`). Ganti *`DB_MasterUsername`* dengan `MasterUsername` nilai yang diberikan dalam HA Stack RFC (misalnya,`admin`).

Ganti *DB_MasterUserPassword* dengan `MasterUserPassword` nilai yang diberikan dalam HA Stack RFC (misalnya,`p4ssw0rd`).
Ganti *DB_ENDPOINT* dengan nama DNS endpoint dalam output eksekusi HA Stack RFC (misalnya,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Anda dapat menemukannya dengan GetRfcoperasi (CLI: get-rfc --rfc-id RFC_ID) atau di halaman detail RFC Konsol AMS untuk HA Stack RFC yang sebelumnya Anda kirimkan.

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. Di direktori yang sama buat `install_dependencies.sh` dengan konten berikut:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS diinstal sebagai bagian dari data pengguna saat peluncuran untuk memungkinkan pemeriksaan kesehatan berfungsi sejak awal.

6. Di direktori yang sama buat `start_server.sh` dengan konten berikut:

   - Untuk instance Amazon Linux, gunakan ini:

     ```
     #!/bin/bash
     service httpd start
     ```

   - Untuk instance RHEL, gunakan ini (perintah tambahan adalah kebijakan yang memungkinkan SELINUX menerima): WordPress

```
#!/bin/bash
setsebool -P  httpd_can_network_connect_db 1
setsebool -P  httpd_can_network_connect 1
chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
restorecon -Rv /var/www/html
service httpd start
```

7. Di direktori yang sama buat `stop_server.sh` dengan konten berikut:

```
#!/bin/bash
service httpd stop
```

8. Buat bundel zip.

   Linux:

```
$ cd /tmp/WordPress
$ zip -r wordpress.zip .
```

   Windows: Buka direktori "WordPress" Anda dan pilih semua file dan buat file zip, pastikan untuk menamainya wordpress.zip.

1. Unggah bundel aplikasi ke bucket S3

   Paket harus ada untuk terus menerapkan tumpukan.

   Anda secara otomatis memiliki akses ke instans bucket S3 yang Anda buat. Anda dapat mengaksesnya melalui Bastions Anda (lihat Mengakses Instans), atau melalui konsol S3, dan mengunggah CodeDeploy paket dengan drag-and-drop, atau dengan menjelajah ke dan memilih file.

   Anda juga dapat menggunakan perintah berikut di jendela shell; pastikan Anda memiliki jalur yang benar ke file zip:

```
aws s3 cp wordpress/wordpress.zip s3://BUCKET_NAME/
```

2. Menyebarkan Bundel WordPress CodeDeploy Aplikasi

   PENYEBARAN APLIKASI PENYEBARAN KODE DATA YANG DIPERLUKAN:

- CodeDeployApplicationName: Nama yang Anda berikan pada CodeDeploy aplikasi.
- CodeDeployGroupName: Karena CodeDeploy aplikasi dan grup keduanya dibuat dari nama yang Anda berikan CodeDeploy aplikasi di tumpukan HA RFC, ini adalah nama yang sama dengan CodeDeployApplicationName.
- S3Bucket: Nama yang Anda berikan pada ember S3.
- S3 BundleType dan S3Key: Ini adalah bagian dari bundel WordPress aplikasi yang Anda gunakan.
- VpcId: VPC yang relevan.

a. Pada halaman Create RFC, pilih kategori Deployment, subcategory Applications, item CodeDeploy application, dan operation Deploy dari RFC CT pick list.

b. Pertahankan opsi Basic default, dan atur nilai seperti yang ditunjukkan.

> **Note**
>
> Referensi CodeDeploy aplikasi, grup CodeDeploy penyebaran, bucket S3, dan bundel yang sebelumnya dibuat.

```
Subject:                                    WP-CD-Deploy-RFC
Description:                                DeployWordPress
S3Bucket:                                   BUCKET_NAME
S3Key:                                      wordpress.zip
S3BundleType:                               zip
CodeDeployApplicationName:                  WordPress
CodeDeployDeploymentGroupName:              WordPress
CodeDeployIgnoreApplicationStopFailures:    false
RevisionType:                               S3



VpcId:                                      VPC_ID
Name:                                       WP-CD-Deploy-Op
TimeoutInMinutes:                           60
```

c. Klik Kirim setelah selesai.

Validasi Penerapan Aplikasi

Arahkan ke titik akhir (LoadBalancerCName) penyeimbang beban yang dibuat sebelumnya, dengan jalur yang diterapkan:/. WordPress WordPress Misalnya:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Anda akan melihat halaman seperti ini:

Meruntuhkan Penerapan Ketersediaan Tinggi

Untuk meruntuhkan penerapan, Anda mengirimkan Delete Stack CT terhadap tumpukan HA Two-Tier, dan bucket S3, dan Anda dapat meminta agar snapshot RDS dihapus (mereka dihapus secara otomatis setelah sepuluh hari, tetapi biayanya sedikit sementara di sana). Kumpulkan tumpukan IDs untuk tumpukan HA dan ember S3 lalu ikuti langkah-langkah ini. Lihat Tumpukan | Hapus.

Tutorial Konsol: Menyebarkan Situs Web Tier dan Tie WordPress

Bagian ini menjelaskan cara menerapkan WordPress situs ketersediaan tinggi (HA) ke lingkungan AMS menggunakan konsol AMS. Kumpulan instruksi ini mencakup contoh pembuatan file paket WordPress CodeDeploy -kompatibel (misalnya zip) yang diperlukan. Penyediaan sumber daya mengikuti perintah yang memungkinkan Anda untuk mengikatnya bersama untuk membentuk “tingkatan.”

> **Note**
>
> Panduan penerapan ini dirancang untuk digunakan dengan OS Linux AMZN.
> Parameter variabel penting dinotasikan sebagai*replaceable*; Namun, Anda mungkin ingin memodifikasi parameter lain agar sesuai dengan situasi Anda.

Ringkasan tugas dan diperlukan RFCs:

1. Buat infrastruktur:
   a. Buat kluster database MySQL RDS
   b. Membuat penyeimbang beban
   c. Buat grup penskalaan Otomatis dan ikat ke penyeimbang beban

   d. Buat bucket S3 untuk aplikasi CodeDeploy
2. Buat bundel WordPress aplikasi (tidak memerlukan RFC)
3. Terapkan bundel WordPress aplikasi dengan CodeDeploy:
   a. Buat CodeDeploy aplikasi
   b. Buat CodeDeploy grup penyebaran
   c. Unggah bundel WordPress aplikasi Anda ke bucket S3 (tidak memerlukan RFC)
   d. Menyebarkan aplikasi CodeDeploy
4. Validasi penerapan
5. Meruntuhkan penyebaran

Deskripsi untuk semua opsi CT, termasuk ChangeTypeId dapat ditemukan di Referensi Jenis Perubahan AMS.

Membuat RFC menggunakan Konsol (Dasar-Dasar)

Ini adalah beberapa langkah yang harus Anda ikuti setiap kali Anda membuat RFC menggunakan Konsol.

1. Klik RFCsdi panel navigasi kiri untuk membuka halaman RFCs daftar, lalu klik Buat RFC.

   Halaman Create RFC terbuka.
2. Pilih salah satu Jelajahi jenis perubahan (default) atau Pilih berdasarkan kategori.
3. Jelajahi jenis perubahan:

   a. Klik opsi buat cepat untuk memulai RFC dengan salah satu jenis perubahan yang paling sering digunakan.

      Area konfigurasi Umum untuk jenis perubahan itu terbuka, baris subjek diisi. Untuk melihat detail jenis perubahan, buka area di bagian atas halaman.

   b. Gunakan area Semua jenis perubahan.

      Filter, alihkan antara tampilan kartu atau tabel, atau urutkan jenis perubahan. Ketika Anda menemukan yang Anda inginkan, pilih dan klik Buat RFC di bagian atas halaman.

      Area konfigurasi Umum untuk jenis perubahan itu terbuka, baris subjek diisi. Untuk melihat detail jenis perubahan, buka area di bagian atas halaman.

4. Pilih berdasarkan kategori:

   a. Pilih Kategori, Subkategori, Item, dan Operasi yang sesuai.

      Kotak detail jenis perubahan muncul di bagian bawah halaman.

   b. Klik Buat RFC di bagian bawah halaman.

   c. Area konfigurasi Umum untuk jenis perubahan itu terbuka, baris subjek diisi. Untuk melihat detail jenis perubahan, buka area di bagian atas halaman.

5. Untuk memastikan orang-orang tertentu mendapatkan pemberitahuan tentang kemajuan RFC, isi alamat Email. Untuk menambahkan detail tentang jenis perubahan, isi Deskripsi. Buka area konfigurasi tambahan untuk menambahkan lebih spesifik tentang RFC.
6. Untuk Penjadwalan pilih Jalankan perubahan ini secepatnya atau Jadwalkan perubahan ini. Jika Anda memilih Jalankan perubahan ini secepatnya, RFC Anda akan dieksekusi segera setelah persetujuan berlalu. Jika Anda memilih Jadwalkan jenis perubahan ini, kalender pilih, waktu, dan zona waktu, akan muncul dan RFC Anda dimulai, setelah pengiriman, sesuai jadwal.
7. Di area konfigurasi Eksekusi, konfigurasikan parameter jenis perubahan. Untuk melihat parameter opsional, buka area konfigurasi tambahan.
8. Saat siap, klik Jalankan.

Menciptakan Infrastruktur

Masuk ke AWS Console untuk akun AMS target dan kemudian Konsol AMS untuk akun tersebut.

Prosedur berikut menjelaskan pembuatan database RDS, penyeimbang beban, dan grup Auto Scaling sedemikian rupa sehingga Anda menggunakan IDs sumber daya untuk membangun infrastruktur.

Buat RDS Stack

Lihat tumpukan RDS | Buat.

Buat Tumpukan ELB

Luncurkan ELB publik.

DATA YANG DIBUTUHKAN:

- `VpcId`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.

- `ELBSubnetIds`: Sebuah array subnet di mana penyeimbang beban akan mendistribusikan lalu lintas. Pilih subnet publik atau pribadi. Temukan Subnet IDs dengan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (CLI list-subnet-summaries:) atau di halaman AMS Console -> VPC details. VPCs
- `VpcId`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.

1. Pada halaman Create RFC, pilih kategori Deployment, subkategori Advanced Stack Components, item Load balancer (ELB) stack, dan klik Create. Pilih Advanced dan terima semua default (termasuk yang tidak memiliki nilai) kecuali yang ditampilkan berikutnya.

```
Subject:                          WP-ELB-RFC
ELBSubnetIds:                     PUBLIC_AZ1
                                PUBLIC_AZ2
ELBScheme                         true
ELBCookieExpirationPeriod         600
VpcId:                            VPC_ID
Name:                             WP-Public-ELB
```

2. Klik Kirim setelah selesai.

Buat Tumpukan Grup Auto Scaling

Luncurkan grup penskalaan Otomatis.

DATA YANG DIBUTUHKAN:

- `VpcId`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.
- `AMI-ID`: Nilai ini menentukan jenis EC2 instance grup Auto Scaling (ASG) Anda yang akan berputar. Pastikan untuk memilih AMI di akun Anda yang dimulai dengan “pelanggan-” dan merupakan sistem operasi yang Anda inginkan. Temukan AMI IDs dengan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (CLI: list-amis) atau di halaman AMS Console -> detail. VPCs VPCs Panduan ini untuk ASGs dikonfigurasi untuk menggunakan AMI Linux.
- `ASGLoadBalancerNames`: Penyeimbang beban yang sebelumnya Anda buat - temukan namanya dengan melihat EC2 Console -> Load Balancers (di navigasi kiri). Perhatikan bahwa ini bukan “Nama” yang Anda tentukan saat Anda membuat ELB sebelumnya.

1. Pada halaman Buat RFC, pilih kategori Deployment, subkategori Advanced Stack Components, item Grup penskalaan otomatis, dan klik Buat. Pilih Advanced dan terima semua default (termasuk yang tidak memiliki nilai) kecuali yang ditampilkan berikutnya.

> **Note**
>
> Tentukan AMS AMI terbaru. Tentukan ELB yang dibuat sebelumnya.

```
Subject:                        WP-ASG-RFC
ASGSubnetIds:                   PRIVATE_AZ1
                                                    PRIVATE_AZ2
ASGAmiId:                       AMI_ID
VpcId:                          VPC_ID
Name:                           WP_ASG
ASGLoadBalancerNames:           ELB_NAME
ASGUserData:
#!/bin/bash
REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed
 's/[a-z]$//')
yum -y install ruby httpd
chkconfig httpd on
service httpd start
touch /var/www/html/status
cd /tmp
curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/install
chmod +x ./install
./install auto
chkconfig codedeploy-agent on
service codedeploy-agent start
```

2. Klik Kirim setelah selesai.

Buat Stack S3

Luncurkan ember S3. Bucket S3 adalah tempat Anda mengunggah bundel aplikasi yang Anda buat.

DATA YANG DIBUTUHKAN:

- `VPC-ID`: Nilai ini menentukan di mana Bucket S3 Anda akan berada, ini harus sama dengan VPC yang digunakan sebelumnya.
- `AccessControl`: Opsi AccessControl daftar pra-set (ACL) adalah`Private`, dan. `PublicRead` Untuk informasi selengkapnya, lihat Amazon Simple Storage Service Canned ACL.
- `BucketName`: Nilai ini menetapkan nama Bucket S3, Anda menggunakannya untuk mengunggah bundel aplikasi Anda. Itu harus unik di seluruh wilayah akun dan tidak dapat menyertakan huruf besar. Menyertakan ID akun Anda sebagai bagian dari BucketName bukan persyaratan tetapi membuatnya lebih mudah untuk mengidentifikasi bucket nanti. Untuk melihat nama bucket S3 yang ada di akun, buka Konsol Amazon S3 untuk akun Anda.

1. Pada halaman Create RFC, pilih kategori Deployment, subkategori Advanced Stack Components, item S3 storage, dan klik Create.

   Anda dapat meninggalkan opsi parameter default di Basic untuk menerima default seperti yang dijelaskan. Untuk menetapkan nilai yang berbeda, pilih Advanced.

   > **Note**
   >
   > Bucket yang digunakan dengan jenis perubahan ini memungkinkan read/write akses penuh ke seluruh akun, jenis perubahan baru mungkin diperlukan untuk memungkinkan izin akses yang lebih terbatas.

   ```
   Subject:            S3-Bucket-RFC
   BucketName:         ACCOUNT_ID-codedeploy-bundles
   AccessControl:      Private

   VpcId:              VPC_ID
   Name:               S3BucketForWP
   ```

2. Klik Kirim setelah selesai.

Buat WordPress CodeDeploy Bundel

Bagian ini memberikan contoh pembuatan bundel penerapan aplikasi.

1. Unduh WordPress, ekstrak file dan buat file. /scripts direktori.

Perintah Linux:

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

Windows: Tempel `https://github.com/WordPress/WordPress/archive/master.zip` ke jendela browser dan unduh file zip.

Buat direktori sementara untuk merakit paket.

Linux:

```
mkdir /tmp/WordPress
```

Windows: Buat direktori WordPress "", Anda akan menggunakan jalur direktori nanti.

2. Ekstrak WordPress sumber ke direktori WordPress "" dan buat file. /scripts direktori.

   Linux:

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows: Buka direktori "WordPress" yang Anda buat dan buat direktori “skrip” di sana.

   Jika Anda berada di lingkungan Windows, pastikan untuk mengatur jenis istirahat untuk file skrip ke Unix (LF). Di Notepad ++, ini adalah opsi di kanan bawah jendela.

3. Buat file CodeDeploy appspec.yl, di WordPress direktori (jika menyalin contoh, periksa lekukan, setiap spasi dihitung). PENTING: Pastikan jalur “sumber” benar untuk menyalin WordPress file (dalam hal ini, di WordPress direktori Anda) ke tujuan yang diharapkan (/var/www/html/WordPress). Dalam contoh, file appspec.ymlada di direktori dengan WordPress file, jadi hanya “/” yang diperlukan. Juga, bahkan jika Anda menggunakan RHEL AMI untuk grup Auto Scaling Anda, biarkan baris “os: linux” apa adanya. Contoh file appspec.yml:

   ```
   version: 0.0
   os: linux
   ```

```
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/config_wordpress.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

4. Buat skrip file bash di file. WordPress /scripts direktori.

   Pertama, buat `config_wordpress.sh` dengan konten berikut (jika Anda mau, Anda dapat mengedit file wp-config.php secara langsung).

   > **Note**
   >
   > Ganti *DBName* dengan nilai yang diberikan dalam HA Stack RFC (misalnya,`wordpress`).
   > Ganti *DB_MasterUsername* dengan `MasterUsername` nilai yang diberikan dalam HA Stack RFC (misalnya,`admin`).
   > Ganti *DB_MasterUserPassword* dengan `MasterUserPassword` nilai yang diberikan dalam HA Stack RFC (misalnya,`p4ssw0rd`).
   > Ganti *DB_ENDPOINT* dengan nama DNS endpoint dalam output eksekusi HA Stack RFC (misalnya,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Anda dapat menemukannya dengan GetRfcoperasi (CLI: get-rfc --rfc-id RFC_ID) atau di halaman detail RFC Konsol AMS untuk HA Stack RFC yang sebelumnya Anda kirimkan.

   ```
   #!/bin/bash
   chmod -R 755 /var/www/html/WordPress
   ```

```
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. Di direktori yang sama buat `install_dependencies.sh` dengan konten berikut:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS diinstal sebagai bagian dari data pengguna saat peluncuran untuk memungkinkan pemeriksaan kesehatan bekerja sejak awal.

6. Di direktori yang sama buat `start_server.sh` dengan konten berikut:

   - Untuk instans Amazon Linux, gunakan ini:

   ```
   #!/bin/bash
   service httpd start
   ```

   - Untuk instance RHEL, gunakan ini (perintah tambahan adalah kebijakan yang memungkinkan SELINUX menerima): WordPress

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. Di direktori yang sama buat `stop_server.sh` dengan konten berikut:

```
#!/bin/bash
```

```
service httpd stop
```

8. Buat bundel zip.

   Linux:

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: Buka direktori "WordPress" Anda dan pilih semua file dan buat file zip, pastikan untuk menamainya wordpress.zip.

Menyebarkan Bundel WordPress Aplikasi dengan CodeDeploy

CodeDeploy Ini adalah layanan penerapan AWS yang mengotomatiskan penerapan aplikasi ke instans Amazon. EC2 Bagian dari proses ini melibatkan pembuatan CodeDeploy aplikasi, membuat grup CodeDeploy penyebaran, dan kemudian menyebarkan aplikasi menggunakan. CodeDeploy

Buat CodeDeploy Aplikasi

CodeDeploy Aplikasi ini hanyalah nama atau wadah yang digunakan oleh AWS CodeDeploy untuk memastikan bahwa grup revisi, konfigurasi penerapan, dan penerapan yang benar direferensikan selama penerapan. Konfigurasi penerapan, dalam hal ini, adalah WordPress bundel yang sebelumnya Anda buat.

DATA YANG DIBUTUHKAN:

- `VpcId`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.
- `CodeDeployApplicationName`: Harus unik di akun. Lihat CodeDeploy Konsol untuk memeriksa nama aplikasi yang ada.

1. Buat CodeDeploy Aplikasi untuk WordPress

   Pada halaman Create RFC, pilih kategori Deployment, subcategory Applications, item CodeDeploy application and operation Create dari RFC CT pick list. Pilih Dasar dan atur nilai seperti yang ditunjukkan. Klik Kirim setelah selesai.

   ```
   Subject:                    CD-WP-App-RFC
   CodeDeployApplicationName:  WordPress
   ```

```
VpcId:                          VPC_ID
Name:                           WP-CD-App
```

2. Klik Kirim setelah selesai.

Membuat Grup CodeDeploy Deployment

Buat grup CodeDeploy penyebaran.

Grup CodeDeploy penyebaran mendefinisikan satu set instance individual yang ditargetkan untuk penerapan.

DATA YANG DIBUTUHKAN:

- `VpcId`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.
- `CodeDeployApplicationName`: Gunakan nilai yang Anda buat sebelumnya.
- `CodeDeployAutoScalingGroups`: Gunakan nama grup Auto Scaling yang Anda buat sebelumnya.
- `CodeDeployDeploymentGroupName`: Nama untuk grup penyebaran. Nama ini harus unik untuk setiap aplikasi yang terkait dengan grup penyebaran.
- `CodeDeployServiceRoleArn`: Gunakan rumus yang diberikan dalam contoh.

1. Pada halaman Create RFC, pilih Category Deployment, subcategory Applications, item CodeDeploy deployment group, dan operation Create dari RFC CT pick list. Pilih Advanced dan atur nilai seperti yang ditunjukkan (hanya Subjek yang diperlukan untuk RFC). Klik Kirim setelah selesai.

> **Note**
>
> Referensikan peran CodeDeploy layanan ARN dalam format ini `"arn:aws:iam::085398962942:role/aws-codedeploy-role"` dan gunakan nama grup penskalaan Otomatis yang dibuat sebelumnya untuk “ASG_NAME”.

```
Description:                    Create CodeDeploy Deployment Group for WP
CodeDeployApplicationName:      WordPress
CodeDeployAutoScalingGroups:    ASG_NAME
CodeDeployDeploymentConfigName: CodeDeployDefault.HalfAtATime
```

```
CodeDeployDeploymentGroupName:    WP CD Group
CodeDeployServiceRoleArn:         arn:aws:iam::ACCOUNT_ID:role/aws-codedeploy-role

VpcId:                            VPC_ID
Name:                             WP Deployment Group
```

2. Klik Kirim setelah selesai.

Unggah WordPress Aplikasi

Anda secara otomatis memiliki akses ke instans bucket S3 apa pun yang Anda buat. Anda dapat mengaksesnya melalui Bastions (lihat Mengakses Instans), atau melalui konsol S3, dan mengunggah bundel. CodeDeploy Bundel harus ada untuk terus menerapkan tumpukan. Contoh menggunakan nama bucket yang dibuat sebelumnya.

Anda dapat menggunakan perintah AWS ini untuk zip up bundel:

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

Menyebarkan WordPress Aplikasi dengan CodeDeploy

Menyebarkan CodeDeploy aplikasi.

DATA YANG DIBUTUHKAN:

- `VPC-ID`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.
- `CodeDeployApplicationName`: Gunakan nama untuk CodeDeploy aplikasi yang Anda buat sebelumnya.
- `CodeDeployDeploymentGroupName`: Gunakan nama grup CodeDeploy penyebaran yang Anda buat sebelumnya.
- `S3Location`(tempat Anda mengunggah bundel aplikasi)`S3Bucket`:: BucketName Yang sebelumnya Anda buat, `S3BundleType` dan`S3Key`: Jenis, dan nama, bundel yang Anda letakkan di toko S3 Anda.

1. Menyebarkan Bundel WordPress CodeDeploy Aplikasi

   Pada halaman Create RFC, pilih kategori Deployment, subcategory Applications, item CodeDeploy application, dan operation Deploy dari RFC CT pick list. Pilih Dasar dan atur nilai seperti yang ditunjukkan. Klik Kirim setelah selesai.

> **Note**
>
> Referensi CodeDeploy aplikasi, grup CodeDeploy penyebaran, bucket S3, dan bundel yang sebelumnya dibuat.

```
Subject:                          WP-CD-Deploy-RFC
CodeDeployApplicationName:        WordPress
CodeDeployDeploymentGroupName:    WPCDGroup
RevisionType:                     S3
S3Bucket:                         ACCOUNT_ID-codedeploy-bundles
S3BundleType:                     zip
S3Key:                            wordpress.zip
VpcId:                            VPC_ID
Name:                             WordPress
```

2. Klik Kirim setelah selesai.

Validasi Penerapan Aplikasi

Arahkan ke titik akhir (ELB CName) penyeimbang beban yang dibuat sebelumnya, dengan jalur yang diterapkan:/. WordPress WordPress Misalnya:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Meruntuhkan Penerapan Aplikasi

Untuk meruntuhkan penerapan, Anda mengirimkan Delete Stack CT ke tumpukan database RDS, penyeimbang beban aplikasi, grup Auto Scaling, bucket S3, dan aplikasi dan grup Code Deploy - enam secara keseluruhan. RFCs Selain itu, Anda dapat mengirimkan permintaan layanan untuk snapshot RDS yang akan dihapus (mereka dihapus secara otomatis setelah sepuluh hari, tetapi biayanya sedikit saat berada di sana). Kumpulkan tumpukan IDs untuk semua dan kemudian ikuti langkah-langkah ini. Lihat Tumpukan | Hapus.

Tutorial CLI: Tumpukan Dua Tingkat Ketersediaan Tinggi (Linux/RHEL)

Bagian ini menjelaskan cara menerapkan tumpukan dua tingkat ketersediaan tinggi (HA) ke dalam lingkungan AMS menggunakan AMS CLI.

> **Note**
>
> Panduan penerapan ini telah diuji di lingkungan AMZN Linux dan RHEL.

Ringkasan tugas dan diperlukan RFCs:

1. Buat infrastruktur (tumpukan dua tingkat HA)
2. Buat bucket S3 untuk aplikasi CodeDeploy
3. Buat bundel WordPress aplikasi dan unggah ke bucket S3
4. Menyebarkan aplikasi dengan CodeDeploy
5. Akses WordPress situs dan masuk untuk memvalidasi penerapan

Sebelum Anda Memulai

Deployment | Advanced Stack Components | High Availability Two Tier Stack Advanced | Create CT membuat grup Auto Scaling, load balancer, database, dan CodeDeploy nama aplikasi serta grup deployment (dengan nama yang sama dengan yang Anda berikan pada aplikasi). Untuk informasi tentang CodeDeploy lihat Apa itu CodeDeploy?

Panduan ini menggunakan RFC High Availability Two-Tier Stack (Advanced) yang menyertakan UserData dan juga menjelaskan cara membuat WordPress bundel yang dapat diterapkan. CodeDeploy

Yang `UserData` ditunjukkan dalam contoh mendapatkan metadata instance seperti ID instance, wilayah, dll, dari dalam instance yang sedang berjalan dengan menanyakan layanan metadata EC2 instance yang tersedia di http://169.254.169.254/latest/meta-data/. Baris ini dalam skrip data pengguna:`REGION=$(curl 169.254.169.254/latest/meta-data/placement/availability-zone/ | sed 's/[a-z]$//')`, mengambil nama zona ketersediaan dari layanan meta-data ke variabel $REGION untuk wilayah yang kami dukung, dan menggunakannya

untuk melengkapi URL untuk bucket S3 tempat agen diunduh. CodeDeploy IP 169.254.169.254 hanya dapat dirutekan dalam VPC (semua dapat menanyakan layanan). VPCs Untuk informasi tentang layanan, lihat Metadata Instans dan Data Pengguna. Perhatikan juga bahwa skrip yang UserData dimasukkan sebagai dijalankan sebagai pengguna “root” dan tidak perlu menggunakan perintah “sudo”.

Panduan ini meninggalkan parameter berikut pada nilai default (ditampilkan):

- Grup Auto Scaling:. `Cooldown=300, DesiredCapacity=2, EBSOptimized=false, HealthCheckGracePeriod=600, IAMInstanceProfile=customer-mc-ec2-instance-profile, InstanceDetailedMonitoring=true, InstanceRootVolumeIops=0, InstanceRootVolumeType=standard, InstanceType=m3.medium, MaxInstances=2, MinInstances=2, ScaleDownPolicyCooldown=300, ScaleDownPolicyEvaluationPeriods=4, ScaleDownPolicyPeriod=60, ScaleDownPolicyScalingAdjustment=-1, ScaleDownPolicyStatistic=Average, ScaleDownPolicyThreshold=35, ScaleMetricName=CPUUtilization, ScaleUpPolicyCooldown=60, ScaleUpPolicyEvaluationPeriods=2, ScaleUpPolicyPeriod=60, ScaleUpPolicyScalingAdjustment=2, ScaleUpPolicyStatistic=Average, ScaleUpPolicyThreshold=75`
- Load Balancer:. `HealthCheckInterval=30, HealthCheckTimeout=5`
- Basis data:`BackupRetentionPeriod=7, Backups=true, InstanceType=db.m3.medium, IOPS=0, MultiAZ=true, PreferredBackupWindow=22:00-23:00, PreferredMaintenanceWindow=wed:03:32-wed:04:02, StorageEncrypted=false, StorageEncryptionKey="", StorageType=gp2`.
- Aplikasi:`DeploymentConfigName=CodeDeployDefault.OneAtATime`.
- Ember S3:`AccessControl=Private`.

PENGATURAN TAMBAHAN:

`RequestedStartTime`dan `RequestedEndTime` jika Anda ingin menjadwalkan RFC Anda: Anda dapat menggunakan time.Is untuk menentukan waktu UTC yang benar. Contoh yang diberikan harus disesuaikan dengan tepat. RFC tidak dapat melanjutkan jika waktu mulai telah berlalu. Atau, Anda dapat mengabaikan nilai-nilai tersebut untuk membuat ASAP RFC yang dijalankan segera setelah persetujuan dilewatkan.

> **Note**
>
> Ada banyak parameter yang mungkin Anda pilih untuk diatur secara berbeda dari seperti yang ditunjukkan. Nilai untuk parameter yang ditunjukkan dalam contoh telah diuji tetapi mungkin tidak tepat untuk Anda.

Buat Infrastruktur

Mengumpulkan data berikut sebelum Anda mulai akan membuat penyebaran berjalan lebih cepat.

DATA YANG DIBUTUHKAN HA STACK:

- AutoScalingGroup:
  - `UserData`: Nilai ini disediakan dalam tutorial ini. Ini termasuk perintah untuk mengatur sumber daya untuk CodeDeploy dan memulai CodeDeploy agen.
  - `AMI-ID`: Nilai ini menentukan jenis EC2 instance grup Auto Scaling (ASG) Anda yang akan berputar. Pastikan untuk memilih AMI di akun Anda yang dimulai dengan “pelanggan-” dan merupakan sistem operasi yang Anda inginkan. Temukan AMI IDs dengan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (CLI: list-amis) atau di halaman AMS Console -> detail. VPCs VPCs Panduan ini untuk ASGs dikonfigurasi untuk menggunakan AMI Linux.
- Database:
  - Parameter ini`DBEngine`,`EngineVersion`,, dan `LicenseModel` harus diatur sesuai dengan situasi Anda meskipun nilai yang ditunjukkan dalam contoh telah diuji.
  - Parameter ini`RDSSubnetIds`,`DBName`,`MasterUsername`,, dan `MasterUserPassword` diperlukan saat menerapkan bundel aplikasi. Untuk RDSSubnet Id, gunakan dua subnet Private.
- LoadBalancer:
  - Parameter ini`DBEngine`,`EngineVersion`,, dan `LicenseModel` harus diatur sesuai dengan situasi Anda meskipun nilai yang ditunjukkan dalam contoh telah diuji.
  - `ELBSubnetIds`: Gunakan dua subnet Publik.
- Aplikasi: `ApplicationName` Nilai menetapkan nama CodeDeploy aplikasi dan nama grup CodeDeploy penyebaran. Anda menggunakannya untuk menyebarkan aplikasi Anda. Itu harus unik di akun. Untuk memeriksa CodeDeploy nama akun Anda, lihat CodeDeploy Konsol. Contoh menggunakan "WordPress" tetapi, jika Anda akan menggunakan nilai itu, pastikan itu belum digunakan.

Prosedur ini menggunakan CT stack dua tingkat (lanjutan) ketersediaan tinggi (ct-06mjngx5flwto) dan CT penyimpanan Create S3 (ct-1a68ck03fn98r). Dari akun Anda yang diautentikasi, ikuti langkah-langkah ini di baris perintah.

1. Luncurkan tumpukan infrastruktur.

   a. Keluarkan parameter eksekusi skema JSON untuk CT stack dua tingkat HA ke file di folder Anda saat ini bernama CreateStackParams .json.

   ```
   aws amscm get-change-type-version --change-type-id "ct-06mjngx5flwto"
    --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
    CreateStackParams.json
   ```

   b. Ubah skema. Ganti *variables* yang sesuai. Misalnya, gunakan OS yang Anda inginkan untuk EC2 instance yang akan dibuat ASG. Rekam `ApplicationName` karena Anda akan menggunakannya nanti untuk menyebarkan aplikasi. Perhatikan bahwa Anda dapat menambahkan hingga 50 tag.

   ```
   {
   "Description":      "HA two tier stack for WordPress",
   "Name":             "WordPressStack",
   "TimeoutInMinutes":  360,
   "Tags": [
           {
               "Key": "ApplicationName",
               "Value": "WordPress"
           }
       ],
   "AutoScalingGroup": {
               "AmiId":    "AMI-ID",
               "UserData": "#!/bin/bash \n
               REGION=$(curl 169.254.169.254/latest/meta-data/placement/
   availability-zone/ | sed 's/[a-z]$//') \n
               yum -y install ruby httpd \n
               chkconfig httpd on \n
               service httpd start \n
               touch /var/www/html/status \n
               cd /tmp \n
               curl -O https://aws-codedeploy-$REGION.s3.amazonaws.com/latest/
   install \n
               chmod +x ./install \n
               ./install auto \n
   ```

```
            chkconfig codedeploy-agent on \n
            service codedeploy-agent start"
    },
    "LoadBalancer": {
        "Public":               true,
        "HealthCheckTarget":    "HTTP:80/status"
    },
    "Database":     {
        "DBEngine":             "MySQL",
        "DBName":               "wordpress",
        "EngineVersion":        "8.0.16 ",
        "LicenseModel":         "general-public-license",
        "MasterUsername":       "admin",
        "MasterUserPassword":   "p4ssw0rd"
    },
    "Application":  {
    "ApplicationName":  "WordPress"
        }
}
```

c. Keluarkan template CreateRfc JSON ke file di folder Anda saat ini bernama CreateStackRfc .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateStackRfc.json
```

d. Ubah template RFC sebagai berikut dan simpan, Anda dapat menghapus dan mengganti isinya. Perhatikan bahwa `RequestedStartTime` dan `RequestedEndTime` sekarang opsional; mengecualikan mereka membuat ASAP RFC yang mengeksekusi segera setelah disetujui (yang biasanya terjadi secara otomatis). Untuk mengirimkan RFC terjadwal, tambahkan nilai-nilai tersebut.

```
{
"ChangeTypeVersion":    "3.0",
"ChangeTypeId":         "ct-06mjngx5flwto",
"Title":                "HA-Stack-For-WP-RFC"
}
```

e. Buat RFC, tentukan CreateStackRfc file.json dan file parameter eksekusi CreateStackParams .json:

```
aws amscm create-rfc --cli-input-json file://CreateStackRfc.json --execution-
parameters file://CreateStackParams.json
```

Anda menerima ID RFC dalam tanggapan. Simpan ID untuk langkah selanjutnya.

f. Kirim RFC:

```
aws amscm submit-rfc --rfc-id  RFC_ID
```

Jika RFC berhasil, Anda tidak menerima output.

g. Untuk memeriksa status RFC, jalankan

```
aws amscm get-rfc --rfc-id RFC_ID
```

Catat ID RFC.

2. Luncurkan ember S3

Mengumpulkan data berikut sebelum Anda mulai akan membuat penyebaran berjalan lebih cepat.

EMBER DATA S3 YANG DIBUTUHKAN:

- `VPC-ID`: Nilai ini menentukan di mana Bucket S3 Anda akan berada. Gunakan ID VPC yang sama dengan yang Anda gunakan sebelumnya.
- `BucketName`: Nilai ini menetapkan nama Bucket S3, Anda menggunakannya untuk mengunggah bundel aplikasi Anda. Itu harus unik di seluruh wilayah akun dan tidak dapat menyertakan huruf besar. Menyertakan ID akun Anda sebagai bagian dari BucketName bukan persyaratan tetapi membuatnya lebih mudah untuk mengidentifikasi bucket nanti. Untuk melihat nama bucket S3 yang ada di akun, buka Konsol Amazon S3 untuk akun Anda.

a. Output parameter eksekusi skema JSON untuk penyimpanan S3 create CT ke file JSON bernama creates3 .json. StoreParams

```
aws amscm get-change-type-version --change-type-id "ct-1a68ck03fn98r"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateS3StoreParams.json
```

b. Ubah skema sebagai berikut, Anda dapat menghapus dan mengganti isinya. Ganti *VPC_ID* dengan tepat. Nilai-nilai dalam contoh telah diuji, tetapi mungkin tidak tepat untuk Anda.

> **Tip**
>
> `BucketName`Harus unik di seluruh wilayah akun dan tidak dapat menyertakan huruf besar. Menyertakan ID akun Anda sebagai bagian dari BucketName bukan persyaratan tetapi membuatnya lebih mudah untuk mengidentifikasi bucket nanti. Untuk melihat nama bucket S3 yang ada di akun, buka Konsol Amazon S3 untuk akun Anda.

```
{
"Description":      "S3BucketForWordPressBundle",
"VpcId":            "VPC_ID",
"StackTemplateId":  "stm-s2b72beb000000000",
"Name":             "S3BucketForWP",
"TimeoutInMinutes":  60,
"Parameters":   {
    "AccessControl":    "Private",
    "BucketName":       "ACCOUNT_ID-BUCKET_NAME"
    }
}
```

c. Keluarkan template JSON CreateRfc ke file, di folder Anda saat ini, bernama StoreRfc createS3 .json:

```
aws amscm create-rfc --generate-cli-skeleton > CreateS3StoreRfc.json
```

d. Ubah dan simpan file creates3 StoreRfc .json, Anda dapat menghapus dan mengganti isinya. Perhatikan bahwa `RequestedStartTime` dan `RequestedEndTime` sekarang opsional; mengecualikan mereka membuat ASAP RFC yang mengeksekusi segera setelah disetujui (yang biasanya terjadi secara otomatis). Untuk mengirimkan RFC terjadwal, tambahkan nilai-nilai tersebut.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-1a68ck03fn98r",
"Title":                "S3-Stack-For-WP-RFC"
```

```
}
```

e. Buat RFC, tentukan file creates3 .json dan file parameter eksekusi creates3 StoreRfc .json: StoreParams

```
aws amscm create-rfc --cli-input-json file://CreateS3StoreRfc.json  --
execution-parameters file://CreateS3StoreParams.json
```

Anda menerima RFC baru sebagai tanggapan. RfcId Simpan ID untuk langkah selanjutnya.

f. Kirim RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Jika RFC berhasil, Anda tidak menerima output.

g. Untuk memeriksa status RFC, jalankan

```
aws amscm get-rfc --rfc-id RFC_ID
```

Membuat, Mengunggah, dan Menyebarkan Aplikasi

Pertama, buat bundel WordPress aplikasi, lalu gunakan CodeDeploy CTs untuk membuat dan menyebarkan aplikasi.

1. Unduh WordPress, ekstrak file dan buat file. /scripts direktori.

   Perintah Linux:

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

   Windows: Tempel `https://github.com/WordPress/WordPress/archive/master.zip` ke jendela browser dan unduh file zip.

   Buat direktori sementara untuk merakit paket.

   Linux:

   ```
   mkdir /tmp/WordPress
   ```

Windows: Buat direktori WordPress "", Anda akan menggunakan jalur direktori nanti.

2. Ekstrak WordPress sumber ke direktori WordPress "" dan buat file. /scripts direktori.

   Linux:

   ```
   unzip master.zip -d /tmp/WordPress_Temp
   cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
   rm -rf /tmp/WordPress_Temp
   rm -f master
   cd /tmp/WordPress
   mkdir scripts
   ```

   Windows: Buka direktori "WordPress" yang Anda buat dan buat direktori “skrip” di sana.

   Jika Anda berada di lingkungan Windows, pastikan untuk mengatur jenis istirahat untuk file skrip ke Unix (LF). Di Notepad ++, ini adalah opsi di kanan bawah jendela.

3. Buat file CodeDeploy appspec.yl, di WordPress direktori (jika menyalin contoh, periksa lekukan, setiap spasi dihitung). PENTING: Pastikan jalur “sumber” benar untuk menyalin WordPress file (dalam hal ini, di WordPress direktori Anda) ke tujuan yang diharapkan (/var/www/html/WordPress). Dalam contoh, file appspec.ymlada di direktori dengan WordPress file, jadi hanya “/” yang diperlukan. Juga, bahkan jika Anda menggunakan RHEL AMI untuk grup Auto Scaling Anda, biarkan baris “os: linux” apa adanya. Contoh file appspec.yml:

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

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

4. Buat skrip file bash di file. WordPress /scripts direktori.

   Pertama, buat `config_wordpress.sh` dengan konten berikut (jika Anda mau, Anda dapat mengedit file wp-config.php secara langsung).

   > **Note**
   >
   > Ganti *DBName* dengan nilai yang diberikan dalam HA Stack RFC (misalnya,`wordpress`).
   > Ganti *DB_MasterUsername* dengan `MasterUsername` nilai yang diberikan dalam HA Stack RFC (misalnya,`admin`).
   > Ganti *DB_MasterUserPassword* dengan `MasterUserPassword` nilai yang diberikan dalam HA Stack RFC (misalnya,`p4ssw0rd`).
   > Ganti *DB_ENDPOINT* dengan nama DNS endpoint dalam output eksekusi HA Stack RFC (misalnya,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Anda dapat menemukannya dengan GetRfcoperasi (CLI: get-rfc --rfc-id RFC_ID) atau di halaman detail RFC Konsol AMS untuk HA Stack RFC yang sebelumnya Anda kirimkan.

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. Di direktori yang sama buat `install_dependencies.sh` dengan konten berikut:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
```

```
service httpd restart
```

> **Note**
>
> HTTPS diinstal sebagai bagian dari data pengguna saat peluncuran untuk memungkinkan pemeriksaan kesehatan berfungsi sejak awal.

6. Di direktori yang sama buat `start_server.sh` dengan konten berikut:

   - Untuk instance Amazon Linux, gunakan ini:

   ```
   #!/bin/bash
   service httpd start
   ```

   - Untuk instance RHEL, gunakan ini (perintah tambahan adalah kebijakan yang memungkinkan SELINUX menerima): WordPress

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. Di direktori yang sama buat `stop_server.sh` dengan konten berikut:

   ```
   #!/bin/bash
   service httpd stop
   ```

8. Buat bundel zip.

   Linux:

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: Buka direktori "WordPress" Anda dan pilih semua file dan buat file zip, pastikan untuk menamainya wordpress.zip.

1. Unggah bundel aplikasi ke bucket S3.

Bundel harus ada untuk terus menerapkan tumpukan.

Anda secara otomatis memiliki akses ke instans bucket S3 yang Anda buat. Anda dapat mengaksesnya melalui benteng Anda, atau melalui konsol S3, dan mengunggah WordPress bundel dengan drag-and-drop atau menjelajah ke dan memilih file zip.

Anda juga dapat menggunakan perintah berikut di jendela shell; pastikan Anda memiliki jalur yang benar ke file zip:

```
aws s3 cp wordpress.zip s3://BUCKET_NAME/
```

2. Menyebarkan bundel WordPress aplikasi.

Mengumpulkan data berikut sebelum Anda mulai akan membuat penyebaran berjalan lebih cepat.

DATA YANG DIBUTUHKAN:

- `VPC-ID`: Nilai ini menentukan di mana Bucket S3 Anda akan berada. Gunakan ID VPC yang sama dengan yang Anda gunakan sebelumnya.
- `CodeDeployApplicationNamedanCodeDeployApplicationName`: `ApplicationName` Nilai yang Anda gunakan dalam HA 2-Tier Stack RFC mengatur CodeDeployApplicationName dan. CodeDeployDeploymentGroupName Contoh menggunakan "WordPress" tetapi Anda mungkin telah menggunakan nilai yang berbeda.
- `S3Location`: Untuk`S3Bucket`, gunakan `BucketName` yang Anda buat sebelumnya. Itu `S3BundleType` dan `S3Key` berasal dari bundel yang Anda letakkan di toko S3 Anda.

a. Output parameter eksekusi skema JSON untuk CodeDeploy aplikasi menyebarkan CT ke file JSON bernama Deploy Params.json. CDApp

```
aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 DeployCDAppParams.json
```

b. Ubah skema sebagai berikut dan simpan sebagai, Anda dapat menghapus dan mengganti isinya.

```
{
```

```
"Description":                          "DeployWPCDApp",
"VpcId":                                "VPC_ID",
"Name":                                 "WordPressCDAppDeploy",
"TimeoutInMinutes":                     60,
"Parameters":   {
    "CodeDeployApplicationName":                "WordPress",
    "CodeDeployDeploymentGroupName":            "WordPress",
    "CodeDeployIgnoreApplicationStopFailures":   false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket":     "BUCKET_NAME",
        "S3BundleType": "zip",
        "S3Key":        "wordpress.zip" }
        }
    }
}
```

c. Keluarkan template JSON CreateRfc ke file, di folder Anda saat ini, bernama Deploy CDApp RFC.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

d. Ubah dan simpan file Deploy CDApp RFC.json, Anda dapat menghapus dan mengganti isinya. Perhatikan bahwa `RequestedStartTime` dan `RequestedEndTime` sekarang opsional; mengecualikan mereka membuat ASAP RFC yang mengeksekusi segera setelah disetujui (yang biasanya terjadi secara otomatis). Untuk mengirimkan RFC terjadwal, tambahkan nilai-nilai tersebut.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-RFC"
}
```

e. Buat RFC, tentukan file Deploy CDApp Rfc dan file parameter eksekusi Deploy CDApp Params:

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

Anda menerima RFC baru sebagai tanggapan. RfcId Simpan ID untuk langkah selanjutnya.

f. Kirim RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Jika RFC berhasil, Anda tidak menerima output.

g. Untuk memeriksa status RFC, jalankan

```
aws amscm get-rfc --rfc-id RFC_ID
```

Validasi Penerapan Aplikasi

Arahkan ke titik akhir (ELB CName) penyeimbang beban yang dibuat sebelumnya, dengan jalur yang diterapkan:/. WordPress WordPress Misalnya:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Meruntuhkan Penerapan Aplikasi

Setelah Anda selesai dengan tutorial, Anda akan ingin meruntuhkan penyebaran sehingga Anda tidak dikenakan biaya untuk sumber daya.

Berikut ini adalah operasi penghapusan tumpukan generik. Anda akan ingin mengirimkannya dua kali, sekali untuk tumpukan HA 2-Tier dan sekali untuk tumpukan bucket S3. Sebagai tindak lanjut terakhir, kirimkan permintaan layanan agar semua snapshot untuk bucket S3 (termasuk ID tumpukan bucket S3 dalam permintaan layanan) dihapus. Mereka secara otomatis dihapus setelah 10 hari, tetapi menghapusnya lebih awal menghemat sedikit biaya.

Panduan ini memberikan contoh penggunaan konsol AMS untuk menghapus tumpukan S3; prosedur ini berlaku untuk menghapus tumpukan apa pun menggunakan konsol AMS.

> **Note**
>
> Jika menghapus ember S3, itu harus dikosongkan dari objek terlebih dahulu.

DATA YANG DIBUTUHKAN:

- `StackId`: Tumpukan untuk digunakan. Anda dapat menemukannya dengan melihat halaman AMS Console Stacks, tersedia melalui tautan di navigasi kiri. Menggunakan AMS SKMS API/CLI, jalankan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (di CLI). `list-stack-summaries`
- ID tipe perubahan untuk panduan ini adalah`ct-0q0bic0ywqk6c`, versinya adalah “1.0", untuk mengetahui versi terbaru, jalankan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

BUAT SEBARIS:

- Keluarkan perintah buat RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris). E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- Kirim RFC menggunakan ID RFC yang dikembalikan dalam operasi create RFC. Sampai diserahkan, RFC tetap di `Editing` negara bagian dan tidak ditindaklanjuti.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- Pantau status RFC dan lihat output eksekusi:

```
aws amscm get-rfc --rfc-id RFC_ID
```

TEMPLATE MEMBUAT:

1. Keluarkan template RFC ke file di folder Anda saat ini; contoh menamainya DeleteStackRfc .json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > DeleteStackRfc.json
   ```

2. Ubah dan simpan DeleteStackRfc file.json. Karena menghapus tumpukan hanya memiliki satu parameter eksekusi, parameter eksekusi dapat berada di DeleteStackRfc file.json itu sendiri (tidak perlu membuat file JSON terpisah dengan parameter eksekusi).

Tanda kutip internal dalam ekstensi ExecutionParameters JSON harus diloloskan dengan garis miring terbalik (\). Contoh tanpa waktu mulai dan berakhir:

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
}
```

3. Buat RFC:

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

Anda menerima RFC baru sebagai tanggapan. RfcId Misalnya:

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

Simpan ID untuk langkah selanjutnya.

4. Kirim RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Jika RFC berhasil, Anda tidak menerima konfirmasi di baris perintah.

5. Untuk memantau status permintaan dan untuk melihat Output Eksekusi:

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

Tutorial CLI: Menyebarkan Situs Web Tier dan Tie WordPress

Bagian ini menjelaskan cara menerapkan WordPress situs ketersediaan tinggi (HA) ke lingkungan AMS menggunakan AMS CLI. Kumpulan instruksi ini mencakup contoh pembuatan file paket WordPress CodeDeploy -kompatibel (misalnya zip) yang diperlukan.

> **Note**
>
> Panduan penerapan ini dirancang untuk digunakan dengan lingkungan AMZN Linux. Parameter variabel penting dinotasikan sebagai*replaceable*; Namun, Anda mungkin ingin memodifikasi parameter lain agar sesuai dengan situasi Anda.

Ringkasan tugas dan diperlukan RFCs:

1. Buat infrastruktur:
   a. Buat RDS Stack (CLI)
   b. Membuat penyeimbang beban
   c. Buat grup penskalaan Otomatis dan ikat ke penyeimbang beban
   d. Buat bucket S3 untuk aplikasi CodeDeploy
2. Buat bundel WordPress aplikasi (tidak memerlukan RFC)
3. Terapkan bundel WordPress aplikasi dengan CodeDeploy:
   a. Buat CodeDeploy aplikasi
   b. Buat CodeDeploy grup penyebaran
   c. Unggah bundel WordPress aplikasi Anda ke bucket S3 (tidak memerlukan RFC)
   d. Menyebarkan aplikasi CodeDeploy
4. Validasi penerapan
5. Meruntuhkan penyebaran

Ikuti semua langkah di baris perintah dari akun Anda yang diautentikasi.

Membuat RFC menggunakan CLI

Untuk informasi rinci tentang pembuatan RFCs, lihat Membuat RFCs; untuk penjelasan tentang parameter RFC umum, lihat Parameter umum RFC.

Buat Infrastruktur

Prosedur berikut menjelaskan pembuatan database RDS, penyeimbang beban, dan grup Auto Scaling sedemikian rupa sehingga Anda menggunakan IDs sumber daya untuk membangun infrastruktur.

Buat RDS Stack (CLI)

Lihat tumpukan RDS | Buat.

Buat Tumpukan ELB

Luncurkan penyeimbang beban publik (ELB). Lihat Load Balancer (ELB) Stack | Buat.

Buat Tumpukan Grup Auto Scaling

Luncurkan grup penskalaan Otomatis.

Lihat Grup Auto Scaling | Buat.

Buat Toko S3

Luncurkan ember S3. Bucket S3 adalah tempat Anda mengunggah bundel aplikasi yang Anda buat. Lihat Penyimpanan S3 | Buat.

Buat WordPress Application Bundle untuk CodeDeploy

Bagian ini memberikan contoh pembuatan bundel penerapan aplikasi.

1. Unduh WordPress, ekstrak file dan buat file. /scripts direktori.

   Perintah Linux:

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

   Windows: Tempel `https://github.com/WordPress/WordPress/archive/master.zip` ke jendela browser dan unduh file zip.

   Buat direktori sementara untuk merakit paket.

   Linux:

   ```
   mkdir /tmp/WordPress
   ```

   Windows: Buat direktori WordPress "", Anda akan menggunakan jalur direktori nanti.
2. Ekstrak WordPress sumber ke direktori WordPress "" dan buat file. /scripts direktori.

   Linux:

```
unzip master.zip -d /tmp/WordPress_Temp
cp -paf /tmp/WordPress_Temp/WordPress-master/* /tmp/WordPress
rm -rf /tmp/WordPress_Temp
rm -f master
cd /tmp/WordPress
mkdir scripts
```

Windows: Buka direktori "WordPress" yang Anda buat dan buat direktori “skrip” di sana.

Jika Anda berada di lingkungan Windows, pastikan untuk mengatur jenis istirahat untuk file skrip ke Unix (LF). Di Notepad ++, ini adalah opsi di kanan bawah jendela.

3. Buat file CodeDeploy appspec.yl, di WordPress direktori (jika menyalin contoh, periksa lekukan, setiap spasi dihitung). PENTING: Pastikan jalur “sumber” benar untuk menyalin WordPress file (dalam hal ini, di WordPress direktori Anda) ke tujuan yang diharapkan (/var/www/html/WordPress). Dalam contoh, file appspec.ymlada di direktori dengan WordPress file, jadi hanya “/” yang diperlukan. Juga, bahkan jika Anda menggunakan RHEL AMI untuk grup Auto Scaling Anda, biarkan baris “os: linux” apa adanya. Contoh file appspec.yml:

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

4. Buat skrip file bash di file. WordPress /scripts direktori.

   Pertama, buat `config_wordpress.sh` dengan konten berikut (jika Anda mau, Anda dapat mengedit file wp-config.php secara langsung).

   > **Note**
   >
   > Ganti *DBName* dengan nilai yang diberikan dalam HA Stack RFC (misalnya,`wordpress`).
   > Ganti *DB_MasterUsername* dengan `MasterUsername` nilai yang diberikan dalam HA Stack RFC (misalnya,`admin`).
   > Ganti *DB_MasterUserPassword* dengan `MasterUserPassword` nilai yang diberikan dalam HA Stack RFC (misalnya,`p4ssw0rd`).
   > Ganti *DB_ENDPOINT* dengan nama DNS endpoint dalam output eksekusi HA Stack RFC (misalnya,). `srt1cz23n45sfg.clgvd67uvydk.us-east-1.rds.amazonaws.com` Anda dapat menemukannya dengan GetRfcoperasi (CLI: get-rfc --rfc-id RFC_ID) atau di halaman detail RFC Konsol AMS untuk HA Stack RFC yang sebelumnya Anda kirimkan.

```
#!/bin/bash
chmod -R 755 /var/www/html/WordPress
cp /var/www/html/WordPress/wp-config-sample.php /var/www/html/WordPress/wp-
config.php
cd /var/www/html/WordPress
sed -i "s/database_name_here/DBName/g" wp-config.php
sed -i "s/username_here/DB_MasterUsername/g" wp-config.php
sed -i "s/password_here/DB_MasterUserPassword/g" wp-config.php
sed -i "s/localhost/DB_ENDPOINT/g" wp-config.php
```

5. Di direktori yang sama buat `install_dependencies.sh` dengan konten berikut:

```
#!/bin/bash
yum install -y php
yum install -y php-mysql
yum install -y mysql
service httpd restart
```

> **Note**
>
> HTTPS diinstal sebagai bagian dari data pengguna saat peluncuran untuk memungkinkan pemeriksaan kesehatan berfungsi sejak awal.

6. Di direktori yang sama buat `start_server.sh` dengan konten berikut:

   - Untuk instance Amazon Linux, gunakan ini:

   ```
   #!/bin/bash
   service httpd start
   ```

   - Untuk instance RHEL, gunakan ini (perintah tambahan adalah kebijakan yang memungkinkan SELINUX menerima): WordPress

   ```
   #!/bin/bash
   setsebool -P  httpd_can_network_connect_db 1
   setsebool -P  httpd_can_network_connect 1
   chcon -t httpd_sys_rw_content_t /var/www/html/WordPress/wp-content -R
   restorecon -Rv /var/www/html
   service httpd start
   ```

7. Di direktori yang sama buat `stop_server.sh` dengan konten berikut:

   ```
   #!/bin/bash
   service httpd stop
   ```

8. Buat bundel zip.

   Linux:

   ```
   $ cd /tmp/WordPress
   $ zip -r wordpress.zip .
   ```

   Windows: Buka direktori "WordPress" Anda dan pilih semua file dan buat file zip, pastikan untuk menamainya wordpress.zip.

Menyebarkan WordPress Application Bundle dengan CodeDeploy

CodeDeploy Ini adalah layanan penerapan AWS yang mengotomatiskan penerapan aplikasi ke instans Amazon. EC2 Bagian dari proses ini melibatkan pembuatan CodeDeploy aplikasi, membuat grup CodeDeploy penyebaran, dan kemudian menyebarkan aplikasi menggunakan. CodeDeploy

Buat CodeDeploy Aplikasi

CodeDeploy Aplikasi ini hanyalah nama atau wadah yang digunakan oleh AWS CodeDeploy untuk memastikan bahwa grup revisi, konfigurasi penerapan, dan penerapan yang benar direferensikan selama penerapan. Konfigurasi penerapan, dalam hal ini, adalah WordPress bundel yang sebelumnya Anda buat.

DATA YANG DIBUTUHKAN:

- `VpcId`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.
- `CodeDeployApplicationName`: Harus unik di akun. Lihat CodeDeploy Konsol untuk memeriksa nama aplikasi yang ada.
- `ChangeTypeIddanChangeTypeVersion`: ID tipe perubahan untuk panduan ini adalah`ct-0ah3gwb9seqk2`, untuk mengetahui versi terbaru, jalankan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0ah3gwb9seqk2
```

1. Output parameter eksekusi skema JSON untuk CodeDeploy aplikasi CT ke file di folder Anda saat ini; contoh nama itu Buat CDApp Params.json.

```
aws amscm get-change-type-version --change-type-id "ct-0ah3gwb9seqk2" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > CreateCDAppParams.json
```

2. Ubah dan simpan file JSON sebagai berikut; Anda dapat menghapus dan mengganti isinya.

```
{
"Description":                    "Create WordPress CodeDeploy App",
"VpcId":                          "VPC_ID",
"StackTemplateId":                "stm-sft6rv00000000000",
"Name":                           "WordPressCDApp",
"TimeoutInMinutes":               60,
"Parameters":   {
```

```
    "CodeDeployApplicationName":    "WordPressCDApp"
    }
}
```

3. Output template JSON untuk CreateRfc ke file di folder Anda saat ini; contoh nama itu Buat CDApp RFC.json.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDAppRfc.json
```

4. Ubah dan simpan file JSON sebagai berikut; Anda dapat menghapus dan mengganti isinya. Perhatikan bahwa `RequestedStartTime` dan `RequestedEndTime` sekarang opsional; mengecualikan mereka menyebabkan RFC dieksekusi segera setelah disetujui (yang biasanya terjadi secara otomatis). UntukKirim RFC “terjadwal”, tambahkan nilai-nilai tersebut.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0ah3gwb9seqk2",
"Title":                "CD-App-For-WP-Stack-RFC"
}
```

5. Buat RFC, tentukan file Create CDApp Rfc dan file parameter eksekusi:

```
aws amscm create-rfc --cli-input-json file://CreateCDAppRfc.json --execution-
parameters file://CreateCDAppParams.json
```

Anda menerima ID RFC dari RFC baru dalam tanggapan. Simpan ID untuk langkah selanjutnya.

6. Kirim RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Jika RFC berhasil, Anda tidak menerima output.

7. Kirim RFC:

```
aws amscm get-rfc --rfc-id RFC_ID
```

Membuat Grup CodeDeploy Deployment

Buat grup CodeDeploy penyebaran.

Grup CodeDeploy penyebaran mendefinisikan satu set instance individual yang ditargetkan untuk penerapan.

DATA YANG DIBUTUHKAN:

- `VpcId`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.
- `CodeDeployApplicationName`: Gunakan nilai yang Anda buat sebelumnya.
- `CodeDeployAutoScalingGroups`: Gunakan nama grup Auto Scaling yang Anda buat sebelumnya.
- `CodeDeployDeploymentGroupName`: Nama untuk grup penyebaran. Nama ini harus unik untuk setiap aplikasi yang terkait dengan grup penyebaran.
- `CodeDeployServiceRoleArn`: Gunakan rumus yang diberikan dalam contoh.
- `ChangeTypeIddanChangeTypeVersion`: ID tipe perubahan untuk panduan ini adalah`ct-2gd0u847qd9d2`, untuk mengetahui versi terbaru, jalankan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2gd0u847qd9d2
```

1. Output parameter eksekusi skema JSON ke file di folder Anda saat ini; contoh nama itu Buat CDDep GroupParams .json.

```
aws amscm get-change-type-version --change-type-id "ct-2gd0u847qd9d2"
 --query "ChangeTypeVersion.ExecutionInputSchema" --output text >
 CreateCDDepGroupParams.json
```

2. Ubah dan simpan file JSON sebagai berikut; Anda dapat menghapus dan mengganti isinya.

```
{
"Description":                      "CreateWPCDDeploymentGroup",
"VpcId":                            "VPC_ID",
"StackTemplateId":                  "stm-sp9lrk00000000000",
"Name":                             "WordPressCDAppGroup",
"TimeoutInMinutes":                 60,
"Parameters":   {
    "CodeDeployApplicationName":        "WordPressCDApp",
    "CodeDeployAutoScalingGroups":      ["ASG_NAME"],
    "CodeDeployDeploymentConfigName":   "CodeDeployDefault.HalfAtATime",
    "CodeDeployDeploymentGroupName":    "UNIQUE_CDDepGroupNAME",
```

```
    "CodeDeployServiceRoleArn":         "arn:aws:iam::ACCOUNT_ID:role/aws-
codedeploy-role"
    }
}
```

3. Output template JSON untuk CreateRfc ke file di folder Anda saat ini; contoh nama itu Create CDDep GroupRfc .json.

```
aws amscm create-rfc --generate-cli-skeleton > CreateCDDepGroupRfc.json
```

4. Ubah dan simpan file JSON sebagai berikut; Anda dapat menghapus dan mengganti isinya. Perhatikan bahwa `RequestedStartTime` dan `RequestedEndTime` sekarang opsional; mengecualikan mereka menyebabkan RFC dieksekusi segera setelah disetujui (yang biasanya terjadi secara otomatis). Untuk mengirimkan RFC “terjadwal”, tambahkan nilai-nilai tersebut.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2gd0u847qd9d2",
"Title":                "CD-Dep-Group-For-WP-Stack-RFC"
}
```

5. Buat RFC, tentukan CDDep GroupRfc file Buat dan file parameter eksekusi:

```
aws amscm create-rfc --cli-input-json file://CreateCDDepGroupRfc.json --execution-
parameters file://CreateCDDepGroupParams.json
```

   Anda menerima ID RFC dari RFC baru dalam tanggapan. Simpan ID untuk langkah selanjutnya.

6. Kirim RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

   Jika RFC berhasil, Anda tidak menerima output.

7. Periksa status RFC:

```
aws amscm get-rfc --rfc-id RFC_ID
```

Unggah WordPress Aplikasi

Anda secara otomatis memiliki akses ke instans bucket S3 apa pun yang Anda buat. Anda dapat mengaksesnya melalui Bastions (lihat Mengakses Instans), atau melalui konsol S3, dan mengunggah bundel. CodeDeploy Bundel harus ada untuk terus menerapkan tumpukan. Contoh menggunakan nama bucket yang dibuat sebelumnya.

```
aws s3 cp wordpress/wordpress.zip s3://ACCOUNT_ID-codedeploy-bundles/
```

Menyebarkan WordPress Aplikasi dengan CodeDeploy

Menyebarkan CodeDeploy aplikasi.

Setelah Anda memiliki bundel CodeDeploy aplikasi dan grup penyebaran, gunakan RFC ini untuk menyebarkan aplikasi.

DATA YANG DIBUTUHKAN:

- `VPC-ID`: VPC yang Anda gunakan, ini harus sama dengan VPC yang digunakan sebelumnya.
- `CodeDeployApplicationName`: Gunakan nama untuk CodeDeploy aplikasi yang Anda buat sebelumnya.
- `CodeDeployDeploymentGroupName`: Gunakan nama grup CodeDeploy penyebaran yang Anda buat sebelumnya.
- `S3Location`(tempat Anda mengunggah bundel aplikasi)`S3Bucket`:: BucketName Yang sebelumnya Anda buat, `S3BundleType` dan`S3Key`: Jenis, dan nama, bundel yang Anda letakkan di toko S3 Anda.
- `ChangeTypeIddanChangeTypeVersion`: ID tipe perubahan untuk panduan ini adalah`ct-2edc3sd1sqmrb`, untuk mengetahui versi terbaru, jalankan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-2edc3sd1sqmrb
```

1. Output parameter eksekusi skema JSON untuk CT penerapan CodeDeploy aplikasi ke file di folder Anda saat ini; contoh nama itu Deploy Params.json. CDApp

```
aws amscm get-change-type-version --change-type-id "ct-2edc3sd1sqmrb" --query
 "ChangeTypeVersion.ExecutionInputSchema" --output text > DeployCDAppParams.json
```

2. Ubah file JSON sebagai berikut; Anda dapat menghapus dan mengganti isinya. Untuk`S3Bucket`, gunakan `BucketName` yang Anda buat sebelumnya.

```
{
"Description":                        "Deploy WordPress CodeDeploy Application",
"VpcId":                              "VPC_ID",
"Name":                               "WP CodeDeploy Deployment Group",
"TimeoutInMinutes":                   60,
"Parameters":   {
    "CodeDeployApplicationName":         "WordPressCDApp",
    "CodeDeployDeploymentGroupName":     "WordPressCDDepGroup",
    "CodeDeployIgnoreApplicationStopFailures": false,
    "CodeDeployRevision": {
      "RevisionType": "S3",
      "S3Location": {
        "S3Bucket": "ACCOUNT_ID.BUCKET_NAME",
        "S3BundleType": "zip",
        "S3Key": "wordpress.zip" }
        }
    }
}
```

3. Output template JSON untuk CreateRfc ke file di folder Anda saat ini; contoh nama itu Deploy CDApp RFC.json:

```
aws amscm create-rfc --generate-cli-skeleton > DeployCDAppRfc.json
```

4. Ubah dan simpan file Deploy CDApp RFC.json; Anda dapat menghapus dan mengganti isinya.

```
{
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-2edc3sd1sqmrb",
"Title":                "CD-Deploy-For-WP-Stack-RFC",
"RequestedStartTime":   "2017-04-28T22:45:00Z",
"RequestedEndTime":     "2017-04-28T22:45:00Z"
}
```

5. Buat RFC, tentukan file parameter eksekusi dan file Deploy CDApp Rfc:

```
aws amscm create-rfc --cli-input-json file://DeployCDAppRfc.json  --execution-
parameters file://DeployCDAppParams.json
```

Anda menerima RFC baru sebagai tanggapan. RfcId Simpan ID untuk langkah selanjutnya.

6. Kirim RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Jika RFC berhasil, Anda tidak menerima output.

Validasi Penerapan Aplikasi

Arahkan ke titik akhir (ELB CName) penyeimbang beban yang dibuat sebelumnya, dengan jalur yang WordPress diterapkan:/. WordPress Misalnya:

```
http://stack-ID-FOR-ELB.us-east-1.elb.amazonaws.com/WordPress
```

Meruntuhkan Penerapan Aplikasi

Untuk meruntuhkan penerapan, Anda mengirimkan Delete Stack CT ke tumpukan database RDS, penyeimbang beban aplikasi, grup Auto Scaling, bucket S3, dan aplikasi dan grup Code Deploy - enam secara keseluruhan. RFCs Selain itu, Anda dapat mengirimkan permintaan layanan untuk snapshot RDS yang akan dihapus (mereka dihapus secara otomatis setelah sepuluh hari, tetapi biayanya sedikit saat berada di sana). Kumpulkan tumpukan IDs untuk semua dan kemudian ikuti langkah-langkah ini.

Panduan ini memberikan contoh penggunaan konsol AMS untuk menghapus tumpukan S3; prosedur ini berlaku untuk menghapus tumpukan apa pun menggunakan konsol AMS.

> **Note**
>
> Jika menghapus ember S3, itu harus dikosongkan dari objek terlebih dahulu.

DATA YANG DIBUTUHKAN:

- `StackId`: Tumpukan untuk digunakan. Anda dapat menemukannya dengan melihat halaman AMS Console Stacks, tersedia melalui tautan di navigasi kiri. Menggunakan AMS SKMS API/CLI, jalankan referensi Untuk AMS SKMS API, lihat tab Laporan di AWS Artifact Console. operasi (di CLI). `list-stack-summaries`

- ID tipe perubahan untuk panduan ini adalah`ct-0q0bic0ywqk6c`, versinya adalah “1.0", untuk mengetahui versi terbaru, jalankan perintah ini:

```
aws amscm list-change-type-version-summaries --filter
 Attribute=ChangeTypeId,Value=ct-0q0bic0ywqk6c
```

BUAT SEBARIS:

- Keluarkan perintah buat RFC dengan parameter eksekusi yang disediakan sebaris (tanda kutip saat memberikan parameter eksekusi sebaris). E

```
aws amscm create-rfc --change-type-id "ct-0q0bic0ywqk6c" --change-type-version "1.0"
 --title "Delete My Stack" --execution-parameters "{\"StackId\":\"STACK_ID\"}"
```

- Kirim RFC menggunakan ID RFC yang dikembalikan dalam operasi create RFC. Sampai diserahkan, RFC tetap di `Editing` negara bagian dan tidak ditindaklanjuti.

```
aws amscm submit-rfc --rfc-id RFC_ID
```

- Pantau status RFC dan lihat output eksekusi:

```
aws amscm get-rfc --rfc-id RFC_ID
```

TEMPLATE MEMBUAT:

1. Keluarkan template RFC ke file di folder Anda saat ini; contoh menamainya DeleteStackRfc .json:

   ```
   aws amscm create-rfc --generate-cli-skeleton > DeleteStackRfc.json
   ```

2. Ubah dan simpan DeleteStackRfc file.json. Karena menghapus tumpukan hanya memiliki satu parameter eksekusi, parameter eksekusi dapat berada di DeleteStackRfc file.json itu sendiri (tidak perlu membuat file JSON terpisah dengan parameter eksekusi).

   Tanda kutip internal dalam ekstensi ExecutionParameters JSON harus diloloskan dengan garis miring terbalik (\). Contoh tanpa waktu mulai dan berakhir:

   ```
   {
   ```

```
"ChangeTypeVersion":    "1.0",
"ChangeTypeId":         "ct-0q0bic0ywqk6c",
"Title":                "Delete-My-Stack-RFC"
"ExecutionParameters":  "{
        \"StackId\":\"STACK_ID\"}"
}
```

3. Buat RFC:

```
aws amscm create-rfc --cli-input-json file://DeleteStackRfc.json
```

Anda menerima RFC baru sebagai tanggapan. RfcId Misalnya:

```
{
"RfcId": "daaa1867-ffc5-1473-192a-842f6b326102"
}
```

Simpan ID untuk langkah selanjutnya.

4. Kirim RFC:

```
aws amscm submit-rfc --rfc-id RFC_ID
```

Jika RFC berhasil, Anda tidak menerima konfirmasi di baris perintah.

5. Untuk memantau status permintaan dan untuk melihat Output Eksekusi:

```
aws amscm get-rfc --rfc-id RFC_ID --query "Rfc.
{Status:Status.Name,Exec:ExecutionOutput}" --output table
```

Pemeliharaan aplikasi

Setelah infrastruktur diterapkan, memperbaruinya secara konsisten di semua lingkungan AMS Anda, dari QA hingga pementasan hingga produksi, adalah tantangannya.

Bagian ini memberikan gambaran umum tentang proses konsumsi beban kerja AMS dan beberapa contoh metode berbeda yang dapat Anda gunakan untuk memperbarui lapisan infrastruktur cloud Anda.

Strategi pemeliharaan aplikasi

Cara Anda menerapkan aplikasi memengaruhi cara Anda memeliharanya. Bagian ini memberikan beberapa strategi untuk pemeliharaan aplikasi.

Pembaruan lingkungan dapat melibatkan salah satu dari perubahan ini:

- Pembaruan keamanan
- Versi baru aplikasi Anda
- Perubahan konfigurasi aplikasi
- Pembaruan untuk dependensi

> **Note**
>
> Untuk penerapan aplikasi apa pun, apa pun metodenya, selalu ajukan permintaan layanan sebelumnya untuk memberi tahu AMS bahwa Anda akan menerapkan aplikasi.

Contoh Instalasi Aplikasi Immutable vs Mutable

Mutabilitas Instance Komputasi	Metode Instalasi Aplikasi	AMI
bisa berubah	dengan CodeDeploy	Disediakan AMS
	Secara manual	

Mutabilitas Instance Komputasi	Metode Instalasi Aplikasi	AMI
	Dengan Koki atau Boneka, Berbasis Tarik	
	Dengan Ansible atau Garam, Berbasis Dorong	
Tetap	Dengan AMI Emas	Kustom (berdasarkan AMS-disediakan)

Penerapan yang dapat diubah dengan AMI yang diaktifkan CodeDeploy

AWS CodeDeploy adalah layanan yang mengotomatiskan penerapan kode ke instans apa pun, termasuk instans dan EC2 instans Amazon yang berjalan di lokasi. Anda dapat menggunakan CodeDeploy AMS untuk membuat dan menyebarkan CodeDeploy aplikasi. Perhatikan bahwa AMS menyediakan profil instans default untuk CodeDeploy aplikasi.

- Amazon Linux (versi 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

Sebelum Anda menggunakan CodeDeploy untuk pertama kalinya, Anda harus menyelesaikan sejumlah langkah pengaturan:

1. Instal atau tingkatkan AWS CLI
2. Buat Peran Layanan untuk AWS CodeDeploy, Anda menggunakan ARN Peran Layanan dalam penerapan

IDs untuk semua opsi CT dapat ditemukan di Referensi Ubah Jenis.

> **Note**
>
> Saat ini, Anda harus menggunakan penyimpanan Amazon S3 dengan solusi ini.

Langkah-langkah dasar diuraikan di sini dan prosedurnya dirinci dalam Panduan Pengguna AMS.

1. Buat ember penyimpanan Amazon S3. CT: ct-1a68ck03fn98r. Bucket S3 harus mengaktifkan versi (untuk informasi tentang hal ini, lihat Mengaktifkan Pembuatan Versi Bucket).
2. Letakkan CodeDeploy artefak Anda yang dibundel di atasnya. Anda dapat melakukan ini dengan konsol Amazon S3 tanpa meminta akses melalui AMS. Atau menggunakan variasi dari perintah ini:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Temukan AMS `customer-` AMI; gunakan:

   - Konsol AMS: Halaman detail VPC untuk VPC yang relevan
   - AMS API Untuk referensi AMS SKMS API, lihat tab Laporan di AWS Artifact Console. atau CLI: `aws amsskms list-amis`

4. Buat grup Autoscaling (ASG). CT: ct-2tylseo8rxfsc. Tentukan AMS AMI, atur penyeimbang beban agar memiliki port terbuka, tentukan `customer-mc-ec2-instance-profile` untuk. `ASGIAMInstanceProfile`
5. Buat CodeDeploy aplikasi Anda. CT: ct-0ah3gwb9seqk2. Parameter termasuk nama aplikasi; misalnya`WordpressProd`.
6. Buat grup CodeDeploy penyebaran Anda. CT: ct-2gd0u847qd9d2. Parameter termasuk nama CodeDeploy aplikasi Anda, nama ASG, nama tipe konfigurasi, dan ARN peran layanan.
7. Menyebarkan CodeDeploy aplikasi. CT: ct-2edc3sd1sqmrb. Parameter mencakup nama CodeDeploy aplikasi Anda, nama tipe konfigurasi, nama grup penerapan, jenis revisi, dan lokasi bucket S3 tempat artefak berada. CodeDeploy

Penerapan yang dapat berubah, instans aplikasi yang dikonfigurasi secara manual, dan diperbarui

Strategi penyebaran aplikasi ini adalah pembaruan sederhana dan manual dari instance aplikasi. Ini adalah langkah-langkah dasar.

IDs untuk semua opsi CT dapat ditemukan di Referensi Ubah Jenis.

> **Note**
>
> Saat ini, Anda harus menggunakan penyimpanan Amazon S3 dengan solusi ini.

Langkah-langkah dasar diuraikan di sini; berbagai prosedur dirinci dalam Panduan Pengguna AMS.

1. Buat ember penyimpanan Amazon S3. CT: ct-1a68ck03fn98r. Bucket S3 harus mengaktifkan versi (untuk informasi tentang hal ini, lihat Mengaktifkan Pembuatan Versi Bucket).
2. Letakkan artefak aplikasi yang dibundel di atasnya (semua yang dibutuhkan aplikasi Anda untuk memulai saat boot dan bekerja). Anda dapat melakukan ini dengan konsol Amazon S3 tanpa meminta akses melalui AMS. Atau menggunakan variasi dari perintah ini:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Temukan AMS AMI, semua akan ada CodeDeploy pada mereka. Untuk menemukan “pelanggan-” AMI gunakan:

   - Konsol AMS: Halaman detail VPC untuk VPC yang relevan
   - AMS API Untuk referensi AMS SKMS API, lihat tab Laporan di AWS Artifact Console. atau CLI: `aws amsskms list-amis`

4. Buat EC2 instance dengan AMI itu. CT: ct-14027q0sjyt1h. Tentukan AMS AMI, atur tag `Key=backup, Value=true` dan tentukan `InstanceProfile` parameter `customer-mc-ec2-instance-profile` untuk. Perhatikan ID contoh yang dikembalikan.
5. Minta akses admin ke instance. CT: ct-1dmlg9g1l91h6. Anda akan memerlukan FQDN untuk akun Anda. Jika Anda tidak yakin apa FQDN Anda, Anda dapat menemukannya dengan:

   - Menggunakan AWS Management Console for Directory Services (di bawah tab Security and Identity) Directory Name.
   - Menjalankan salah satu perintah ini (kelas direktori kembali; DC+DC+DC = FQDN): Windows: atau Linux:. `whoami /fqdn` `hostname --fqdn`

6. Masuk ke instans, lihat Mengakses Instans melalui Bastion di Panduan Pengguna AMS.
7. Unduh file aplikasi yang dibundel dari bucket S3 Anda ke instans.
8. Minta cadangan langsung dengan permintaan layanan ke AMS, Anda harus mengetahui ID instans.

9. Saat Anda perlu memperbarui aplikasi Anda, muat file baru ke bucket S3 Anda dan kemudian ikuti langkah 3 hingga 8.

Penerapan yang dapat diubah dengan AMI yang dikonfigurasi alat penerapan berbasis tarik

Strategi ini bergantung pada `InstanceUserData` parameter dalam Managed Services Create EC2 CT. Untuk informasi selengkapnya tentang penggunaan parameter ini, lihat Mengonfigurasi Instans dengan Data Pengguna. Contoh ini mengasumsikan alat penyebaran aplikasi berbasis tarik seperti Chef atau Puppet.

CodeDeploy Agen didukung di semua AMS AMIs. Berikut adalah daftar yang didukung AMIs:

- Amazon Linux (versi 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

IDs untuk semua opsi CT dapat ditemukan di Referensi Ubah Jenis.

> **Note**
>
> Saat ini, Anda harus menggunakan penyimpanan Amazon S3 dengan solusi ini.

Langkah-langkah dasar diuraikan di sini dan prosedurnya dirinci dalam Panduan Pengguna AMS.

1. Buat ember penyimpanan Amazon S3. CT: ct-1a68ck03fn98r. Bucket S3 harus mengaktifkan versi (untuk informasi tentang hal ini, lihat Mengaktifkan Pembuatan Versi Bucket).
2. Letakkan CodeDeploy artefak Anda yang dibundel di atasnya. Anda dapat melakukan ini dengan konsol Amazon S3 tanpa meminta akses melalui AMS. Atau menggunakan variasi dari perintah ini:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Temukan AMS `customer-` AMI; gunakan salah satu:

- Konsol AMS: Halaman detail VPC untuk VPC yang relevan
- AMS API Untuk referensi AMS SKMS API, lihat tab Laporan di AWS Artifact Console. atau CLI: `aws amsskms list-amis`

4. Buat sebuah EC2 instance. CT: ct-14027q0sjyt1h; atur tag`Key=backup, Value=true`, dan gunakan `InstanceUserData` parameter untuk menentukan bootstrap dan skrip lainnya (agen unduhan Chef/Puppet , dll.), Dan sertakan kunci otorisasi yang diperlukan. Anda dapat menemukan contoh melakukan hal ini di Panduan Pengguna AMS, contoh bagian Change Mangement untuk membuat HA Two Tier Deployment. Atau, minta akses ke, dan masuk ke, instance dan konfigurasikan dengan artefak penerapan yang diperlukan. Ingat bahwa perintah penerapan berbasis tarik beralih dari agen pada instance Anda ke server master perusahaan Anda dan mungkin memerlukan otorisasi untuk melewati benteng. Anda mungkin memerlukan permintaan layanan ke AMS untuk meminta akses group/AD grup keamanan tanpa benteng.
5. Ulangi langkah 4 untuk membuat EC2 instance lain dan mengkonfigurasinya dengan server master tool deployment.
6. Saat Anda perlu memperbarui aplikasi, gunakan alat penyebaran untuk meluncurkan pembaruan ke instance Anda.

Penerapan yang dapat diubah dengan AMI yang dikonfigurasi alat penerapan berbasis push

Strategi ini bergantung pada `InstanceUserData` parameter dalam Managed Services Create EC2 CT. Untuk informasi selengkapnya tentang penggunaan parameter ini, lihat Mengonfigurasi Instans dengan Data Pengguna. Contoh ini mengasumsikan alat penyebaran aplikasi berbasis tarik seperti Chef atau Puppet.

IDs untuk semua opsi CT dapat ditemukan di Referensi Ubah Jenis.

> **Note**
>
> Saat ini, Anda harus menggunakan penyimpanan Amazon S3 dengan solusi ini.

Langkah-langkah dasar diuraikan di sini dan prosedurnya dirinci dalam Panduan Pengguna AMS.

1. Buat ember penyimpanan Amazon S3. CT: ct-1a68ck03fn98r. Bucket S3 harus mengaktifkan versi (untuk informasi tentang hal ini, lihat Mengaktifkan Pembuatan Versi Bucket).
2. Letakkan CodeDeploy artefak Anda yang dibundel di atasnya. Anda dapat melakukan ini dengan konsol Amazon S3 tanpa meminta akses melalui AMS. Atau menggunakan variasi dari perintah ini:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Temukan AMS AMI, semua akan ada CodeDeploy pada mereka. Untuk menemukan "pelanggan-" AMI gunakan:

   - Konsol AMS: Halaman detail VPC untuk VPC yang relevan
   - AMS API Untuk referensi AMS SKMS API, lihat tab Laporan di AWS Artifact Console. atau CLI: `aws amsskms list-amis`

4. Buat sebuah EC2 instance. CT: ct-14027q0sjyt1h; setel `tagKey=backup, Value=true,` dan gunakan `InstanceUserData` parameter untuk menjalankan bootstrap dan skrip lainnya termasuk kunci otorisasi, tumpukan SALT (bootstrap minion—untuk informasi lebih lanjut lihat Bootstrapping Salt di Linux dengan EC2 Cloud-Init) atau Ansible (instal pasangan kunci—untuk informasi lebih lanjut lihat Memulai dengan Manajemen Inventaris Amazon Ansible dan Dinamis). EC2 Bergantian, minta akses ke, dan masuk ke, instance dan konfigurasikan dengan artefak penyebaran yang diperlukan. Ingat bahwa perintah berbasis push berasal dari subnet perusahaan Anda ke instance Anda dan Anda mungkin perlu mengonfigurasi otorisasi agar mereka dapat melewati benteng. Anda mungkin memerlukan permintaan layanan ke AMS untuk meminta akses group/AD grup keamanan tanpa benteng.
5. Ulangi langkah 4 untuk membuat EC2 instance lain dan mengkonfigurasinya dengan server master tool deployment.
6. Saat Anda perlu memperbarui aplikasi, gunakan alat penyebaran untuk meluncurkan pembaruan ke instance Anda.

Penerapan yang tidak dapat diubah dengan AMI emas

Strategi ini menggunakan AMI "emas" yang telah Anda konfigurasikan untuk berperilaku seperti yang Anda inginkan untuk semua instance aplikasi Anda. Misalnya, instance yang dibuat dengan AMI emas ini akan bergabung sendiri dengan domain dan DNS yang benar, mengkonfigurasi sendiri, reboot, dan meluncurkan semua sistem yang diperlukan. Ketika Anda ingin memperbarui instance

aplikasi Anda, Anda membuat ulang AMI emas dan meluncurkan semua instance aplikasi baru dengannya.

CodeDeploy Agen didukung di semua AMS AMIs. Berikut adalah daftar yang didukung AMIs:

- Amazon Linux (versi 1)
- Amazon Linux 2
- RedHat 7
- CentOS 7

IDs untuk semua opsi CT dapat ditemukan di Referensi Ubah Jenis.

> **Note**
>
> Saat ini, Anda harus menggunakan penyimpanan Amazon S3 dengan solusi ini.

1. Buat ember penyimpanan Amazon S3. CT: ct-1a68ck03fn98r. Bucket S3 harus mengaktifkan versi (untuk informasi tentang hal ini, lihat Mengaktifkan Pembuatan Versi Bucket).
2. Letakkan artefak aplikasi yang dibundel di atasnya (semua yang dibutuhkan aplikasi Anda untuk memulai saat boot dan bekerja). Anda dapat melakukan ini dengan konsol Amazon S3 tanpa meminta akses melalui AMS. Atau menggunakan variasi dari perintah ini:

   ```
   aws s3 cp ZIP_FILEPATH_AND_NAME s3://S3BUCKET_NAME/
   ```

3. Temukan AMS `customer-` AMI; gunakan salah satu:
   - Konsol AMS: Halaman detail VPC untuk VPC yang relevan
   - AMS API Untuk referensi AMS SKMS API, lihat tab Laporan di AWS Artifact Console. atau CLI: `aws amsskms list-amis`
4. Buat EC2 instance dengan AMI itu. CT: ct-14027q0sjyt1h. Tentukan AMS AMI, tetapkan tag `Key=backup, Value=true` dan tentukan `customer-mc-ec2-instance-profile` untuk`InstanceProfile`. Perhatikan ID contoh yang dikembalikan.
5. Minta akses admin ke instance. CT: ct-1dmlg9g1l91h6. Anda akan memerlukan FQDN untuk akun Anda. Jika Anda tidak yakin apa FQDN Anda, Anda dapat menemukannya dengan:

- Menggunakan AWS Management Console for Directory Services (di bawah tab Security and Identity) Directory Name.
- Menjalankan salah satu perintah ini (kelas direktori kembali; DC+DC+DC = FQDN): Windows: atau Linux:. `whoami /fqdn` `hostname --fqdn`

6. Masuk ke instans, lihat Mengakses Instans di Panduan Pengguna AMS.
7. Unduh ke instance file aplikasi yang dibundel dari bucket S3 Anda. Konfigurasikan instance sehingga dapat menyebarkan aplikasi yang berfungsi penuh saat boot.
8. Buat AMI emas pada instance. CT: ct-3rqqu43krekby. Untuk detailnya, lihat AMI | Buat.
9. Konfigurasikan grup Auto Scaling untuk membuat instance baru menggunakan AMI tersebut. CT: ct-2tylseo8rxfsc. Saat Anda perlu memperbarui aplikasi Anda, ikuti prosedur ini dan minta AMS memperbarui ASG untuk menggunakan AMI emas baru; gunakan Manajemen | Lainnya | Lainnya | Perbarui CT untuk ini.

Perbarui Strategi

Ada beberapa strategi berbeda yang dapat Anda terapkan untuk memperbarui aplikasi atau instans Anda di lingkungan yang dikelola AMS Anda.

- Waktu Henti Terjadwal: Strategi sederhana ini melibatkan waktu penjadwalan agar aplikasi Anda offline dan diperbarui secara manual. Untuk melakukan ini, kirimkan permintaan Manajemen | Lainnya | Lainnya | Perbarui CT (ct-0xdawir96cy7k) untuk menghentikan instance yang diperlukan. Buat pembaruan yang diperlukan, lalu kirimkan Manajemen lain | Lainnya | Lainnya | Perbarui permintaan CT (ct-0xdawir96cy7k) untuk memulai instance.

- Biru/Hijau: Strategi ini mengharuskan Anda memiliki lingkungan yang berlebihan (dua lingkungan yang sepenuhnya fungsional) dan mengambil satu lingkungan offline menggunakan pembaruan sistem nama domain (DNS) atau firewall web (WAF) untuk mengarahkan lalu lintas. Perbarui satu lingkungan dan kemudian alihkan lagi untuk memperbarui lingkungan lainnya.

  Untuk mempelajari lebih lanjut, lihat AWS CodeDeploy Memperkenalkan Blue/Green Penerapan.
- Pembaruan Bergulir dengan AMI baru: Di sinilah Anda memiliki AMI baru yang Anda sesuaikan (lihat Buat AMI) dan kemudian meminta AMS menerapkannya ke grup Auto Scaling Anda. Gunakan Manajemen | Lainnya | Lainnya | Perbarui CT (ct-0xdawir96cy7k) untuk melakukan ini.

AWS Managed Services Resource Scheduler

Gunakan AWS Managed Services (AMS) Resource Scheduler untuk menjadwalkan mulai dan berhenti otomatis AutoScaling grup, EC2 instans Amazon, dan instans RDS di akun Anda. Ini membantu mengurangi biaya infrastruktur di mana sumber daya tidak dimaksudkan untuk berjalan 24/7. Solusinya dibangun di atas Penjadwal Instance AWS, tetapi berisi fitur dan penyesuaian tambahan khusus untuk kebutuhan AMS.

> **Note**
>
> Secara default, Penjadwal Sumber Daya AMS tidak berinteraksi dengan sumber daya yang bukan bagian dari AWS CloudFormation tumpukan. Sumber daya harus menjadi bagian dari tumpukan yang dimulai dengan “stack-”, “sc-” atau “SC-”. Untuk menjadwalkan sumber daya yang bukan bagian dari CloudFormation tumpukan, Anda dapat memperbarui parameter tumpukan Resource Scheduler `ScheduleNonStackResources` ke`Yes`.

Penjadwal Sumber Daya AMS menggunakan periode dan jadwal:

- Periode menentukan waktu saat Resource Scheduler berjalan, seperti waktu mulai, waktu akhir, dan hari dalam sebulan.
- Jadwal berisi periode yang ditentukan, bersama dengan konfigurasi tambahan, seperti jendela pemeliharaan SSM, zona waktu, pengaturan hibernasi, dan sebagainya; dan tentukan kapan sumber daya harus dijalankan, mengingat aturan periode yang dikonfigurasi.

Anda dapat mengonfigurasi periode dan jadwal ini menggunakan tipe perubahan otomatis (CTs) AMS Resource Scheduler.

Untuk detail selengkapnya tentang setelan yang tersedia untuk Penjadwal Sumber Daya AMS, lihat dokumentasi Penjadwal AWS Instance terkait di komponen Solusi. Untuk tampilan arsitektur solusi, lihat dokumentasi Penjadwal AWS Instance yang sesuai di Architecture overview.html.

Menerapkan Penjadwal Sumber Daya AMS

Untuk menerapkan Penjadwal Sumber Daya AMS, gunakan tipe perubahan otomatis (CT): Deployment | AMS Resource Scheduler | Solution | Deploy (ct-0ywnhc8e5k9z5) untuk memunculkan RFC yang kemudian menyebarkan solusi di akun Anda. Setelah RFC dijalankan, CloudFormation

tumpukan yang berisi sumber daya Penjadwal Sumber Daya AMS dengan konfigurasi default, secara otomatis disediakan ke akun Anda. Untuk selengkapnya tentang jenis perubahan Resource Scheduler, lihat Penjadwal Sumber Daya AMS.

> **Note**
>
> Untuk mengetahui apakah Penjadwal Sumber Daya AMS sudah digunakan di akun Anda, periksa konsol AWS Lambda untuk akun tersebut dan cari fungsi Scheduler. AMSResource

Setelah Penjadwal Sumber Daya AMS disediakan di akun Anda, kami sarankan Anda meninjau konfigurasi default dan, jika diperlukan, menyesuaikan konfigurasi seperti kunci tag, zona waktu, layanan terjadwal, dan sebagainya, berdasarkan preferensi Anda. Untuk detail tentang penyesuaian yang disarankan, lihatMenyesuaikan Penjadwal Sumber Daya AMS, selanjutnya.

Untuk membuat konfigurasi kustom, atau hanya mengonfirmasi konfigurasi Resource Scheduler,

Menyesuaikan Penjadwal Sumber Daya AMS

Sebaiknya Anda menyesuaikan properti berikut dari Penjadwal Sumber Daya AMS menggunakan pemutakhiran jenis perubahan Penjadwal Sumber Daya AMS, lihat Penjadwal Sumber Daya AMS.

- Nama tag: Nama tag yang akan digunakan Resource Scheduler untuk mengaitkan jadwal instance dengan sumber daya. Nilai defaultnya adalah Jadwal.
- Layanan Terjadwal: Daftar layanan yang dipisahkan koma yang dapat dikelola oleh Resource Scheduler. Nilai default adalah “ec2, rds, autoscaling”. Nilai yang valid adalah “ec2", “rds” dan “autoscaling”
- Zona waktu default: Tentukan zona waktu default untuk Resource Scheduler yang akan digunakan. Nilai defaultnya adalah UTC.
- Gunakan CMK: Daftar Amazon KMS Customer Managed Key (CMK) yang dipisahkan koma ARNs yang dapat diberikan izin kepada Resource Scheduler.
- Penggunaan LicenseManager: Daftar Manajer AWS Lisensi yang dipisahkan koma ARNs untuk Penjadwal Sumber Daya tersebut dapat diberikan izin untuk.

> **Note**
>
> AMS dapat, dari waktu ke waktu, merilis fitur dan perbaikan agar AMS Resource Scheduler tetap up to date di akun Anda. Ketika ini terjadi, penyesuaian apa pun yang Anda buat pada Penjadwal Sumber Daya AMS dipertahankan.

Menggunakan Penjadwal Sumber Daya AMS

Untuk mengonfigurasi Penjadwal Sumber Daya AMS setelah solusi diterapkan, gunakan Penjadwal Sumber Daya otomatis CTs untuk membuat, menghapus, memperbarui, dan menjelaskan (mendapatkan detail tentang) periode Penjadwal Sumber Daya AMS (waktu saat Resource Scheduler berjalan) dan jadwal (periode yang dikonfigurasi dan opsi lainnya). Untuk contoh penggunaan jenis perubahan Penjadwal Sumber Daya AMS, lihat Penjadwal Sumber Daya AMS.

Untuk memilih sumber daya yang akan dikelola oleh Penjadwal Sumber Daya AMS, setelah penerapan dan pembuatan jadwal, Anda menggunakan AMS Tag Create CTs untuk menandai grup Auto Scaling, tumpukan Amazon RDS, dan sumber daya EC2 Amazon dengan kunci tag yang Anda berikan selama penerapan, dan jadwal yang ditentukan sebagai nilai tag. Setelah sumber daya ditandai, sumber daya dijadwalkan untuk memulai atau berhenti sesuai jadwal Resource Scheduler yang ditentukan.

Tidak ada biaya tambahan untuk menggunakan Penjadwal Sumber Daya AMS. Namun solusinya menggunakan beberapa Layanan AWS dan Anda dikenakan biaya untuk sumber daya ini saat digunakan. Untuk detail selengkapnya, lihat Ikhtisar arsitektur.

Untuk memilih keluar dari Penjadwal Sumber Daya AMS:

- Untuk memilih keluar atau menonaktifkan sementara: Kirim RFC menggunakan Manajemen otomatis | Penjadwal Sumber Daya AMS | Status | Nonaktifkan jenis perubahan (ct-14v49adibs4db)
- Untuk penghapusan permanen: Kirim Manajemen | Lainnya | Lainnya | Pembaruan (diperlukan tinjauan) (ct-0xdawir96cy7k) RFC meminta penghapusan dari sistem otomatisasi rilis Resource Scheduler

Penaksir biaya Penjadwal Sumber Daya AMS

Untuk melacak penghematan biaya, AMS Resource Scheduler menampilkan komponen yang menghitung perkiraan penghematan biaya per jam untuk sumber daya Amazon EC2 dan RDS yang

dikelola oleh penjadwal. Data penghematan biaya ini kemudian diterbitkan sebagai CloudWatch metrik (`AMS/ResourceScheduler`) untuk membantu Anda melacaknya. Estimator penghematan biaya hanya memperkirakan penghematan pada jam kerja instans. Ini tidak memperhitungkan biaya lain, seperti biaya transfer data yang terkait dengan sumber daya.

Estimator penghematan biaya diaktifkan dengan Resource Scheduler. Ini berjalan setiap jam dan mengambil data biaya dan penggunaan dari. AWS Cost Explorer Dari data itu menghitung biaya rata-rata per jam untuk setiap jenis instans dan kemudian memproyeksikan biaya untuk sehari penuh jika berjalan tanpa dijadwalkan. Penghematan biaya adalah perbedaan antara biaya yang diproyeksikan dan biaya yang dilaporkan aktual dari Cost Explorer untuk hari tertentu.

Misalnya, jika instance A dikonfigurasi dengan Resource Scheduler untuk dijalankan dari jam 9 pagi sampai jam 5 sore, yaitu delapan jam pada hari tertentu. Cost Explorer melaporkan biaya sebagai $1 dan penggunaan sebagai 8. Oleh karena itu, biaya rata-rata per jam adalah $0,125. Jika instance tidak dijadwalkan dengan Resource Scheduler, maka instance akan berjalan 24 jam pada hari itu. Dalam hal ini, biayanya adalah 24x0,125 = $3. Resource Scheduler membantu Anda mencapai penghematan biaya $2.

Agar estimator penghematan biaya dapat mengambil biaya dan penggunaan hanya untuk sumber daya yang dikelola oleh Resource Scheduler dari Cost Explorer, kunci tag yang digunakan Resource Scheduler untuk menargetkan sumber daya perlu diaktifkan sebagai tag alokasi Biaya di Dasbor Penagihan. Jika akun milik suatu organisasi, kunci tag harus diaktifkan di akun Manajemen organisasi. Untuk informasi tentang hal ini, lihat Mengaktifkan Tag Alokasi Biaya yang Ditentukan Pengguna dan Tag Alokasi Biaya yang Ditentukan Pengguna

Setelah kunci tag diaktifkan sebagai Tag Alokasi Biaya, AWS penagihan mulai melacak biaya dan penggunaan sumber daya yang dikelola oleh Resource Scheduler, dan setelah data tersebut tersedia, estimator penghematan biaya mulai menghitung penghematan biaya dan mempublikasikan data di bawah namespace metrik di`AMS/ResourceScheduler`. CloudWatch

Kiat penaksir biaya

Estimator Penghematan Biaya tidak menerima diskon seperti instans cadangan, rencana tabungan, dan sebagainya, menjadi pertimbangan dalam perhitungannya. Estimator mengambil biaya penggunaan dari Cost Explorer dan menghitung biaya rata-rata per jam untuk sumber daya. Untuk detail selengkapnya, lihat Memahami Kumpulan Data AWS Biaya Anda: Lembar Cheat

Agar estimator penghematan biaya dapat mengambil biaya dan penggunaan hanya untuk sumber daya yang dikelola oleh Resource Scheduler dari Cost Explorer, kunci tag yang digunakan Resource

Scheduler untuk menargetkan sumber daya perlu diaktifkan sebagai tag Alokasi Biaya di Dasbor Penagihan. Jika akun milik suatu organisasi, kunci tag harus diaktifkan di akun manajemen organisasi. Untuk informasi tentang hal ini, lihat Tag Alokasi Biaya yang Ditentukan Pengguna. Jika tag alokasi biaya tidak diaktifkan, estimator tidak dapat menghitung penghematan dan menerbitkan metrik, bahkan jika itu diaktifkan.

Praktik terbaik Penjadwal Sumber Daya AMS

Menjadwalkan Instans Amazon EC2

- Perilaku mematikan instance harus disetel ke `stop` dan bukan `keterminate`. Ini telah disetel sebelumnya ke `stop` untuk instance yang dibuat dengan jenis perubahan otomatis AMS Amazon EC2 Create (ct-14027q0sjyt1h) dan dapat disetel untuk instans Amazon yang dibuat dengan konsumsi, dengan menyetel properti ke. EC2 AWS CloudFormation `InstanceInitiatedShutdownBehavior` `stop` Jika instance telah mematikan perilaku yang disetel `keterminate`, maka instance akan berakhir ketika Resource Scheduler menghentikannya dan penjadwal tidak akan dapat memulainya kembali.
- EC2 Instans Amazon yang merupakan bagian dari grup Auto Scaling tidak diproses secara individual oleh Penjadwal Sumber Daya AMS, meskipun diberi tag.
- Jika volume root instans target dienkripsi dengan kunci master pelanggan (CMK) KMS, `kms:CreateGrant` izin tambahan perlu ditambahkan ke peran IAM Resource Scheduler Anda, agar penjadwal dapat memulai instance tersebut. Izin ini tidak ditambahkan ke peran secara default untuk meningkatkan keamanan. Jika Anda memerlukan izin ini, kirimkan RFC dengan Management | AMS Resource Scheduler | Solution | Update tipe perubahan, dan tentukan daftar KMS yang dipisahkan koma. ARNs CMKs

Penjadwalan grup Auto Scaling

- Penjadwal Sumber Daya AMS memulai atau menghentikan penskalaan otomatis grup Auto Scaling, bukan instance individual dalam grup. Artinya, penjadwal mengembalikan ukuran grup Auto Scaling (mulai) atau menetapkan ukuran ke 0 (berhenti).
- Tag AutoScaling grup dengan tag yang ditentukan dan bukan instance dalam grup.
- Selama berhenti, Penjadwal Sumber Daya AMS menyimpan nilai kapasitas Minimum, Diinginkan, dan Maksimum grup Auto Scaling dan menetapkan Kapasitas Minimum dan yang Diinginkan ke 0. Selama start, scheduler mengembalikan ukuran grup Auto Scaling seperti saat berhenti. Oleh karena itu, instans grup Auto Scaling harus menggunakan konfigurasi kapasitas yang sesuai

sehingga penghentian dan peluncuran ulang instans tidak memengaruhi aplikasi apa pun yang berjalan di grup Auto Scaling.

- Jika grup Auto Scaling diubah (kapasitas minimum atau maksimum) selama periode berjalan, penjadwal menyimpan ukuran grup Auto Scaling baru dan menggunakannya saat memulihkan grup di akhir jadwal berhenti.

Menjadwalkan instans Amazon RDS

- Penjadwal dapat mengambil snapshot sebelum menghentikan instance RDS (tidak berlaku untuk cluster Aurora DB). Fitur ini diaktifkan secara default dengan parameter CloudFormation template Create RDS Instance Snapshot disetel ke true. Snapshot disimpan hingga saat berikutnya instans Amazon RDS dihentikan dan snapshot baru dibuat.

  Scheduler dapat menggunakan instans start/stop Amazon RDS yang merupakan bagian dari cluster atau database Amazon RDS Aurora atau dalam konfigurasi multi availability zone (Multi-AZ). Namun, periksa batasan Amazon RDS saat penjadwal tidak dapat menghentikan instans Amazon RDS, terutama instans Multi-AZ. Untuk menjadwalkan Aurora Cluster untuk memulai atau berhenti gunakan parameter template Schedule Aurora Clusters (defaultnya benar). Cluster Aurora (bukan instance individual dalam cluster) harus diberi tag dengan kunci tag yang ditentukan selama konfigurasi awal dan nama jadwal sebagai nilai tag untuk menjadwalkan cluster tersebut.

  Setiap instans Amazon RDS memiliki jendela pemeliharaan mingguan di mana setiap perubahan sistem diterapkan. Selama jendela pemeliharaan, Amazon RDS akan secara otomatis memulai instance yang telah dihentikan selama lebih dari tujuh hari untuk menerapkan pemeliharaan. Perhatikan bahwa Amazon RDS tidak akan menghentikan instance setelah acara pemeliharaan selesai.

  Penjadwal memungkinkan menentukan apakah akan menambahkan jendela pemeliharaan yang disukai dari instans Amazon RDS sebagai periode berjalan ke jadwalnya. Solusinya akan memulai instance di awal jendela pemeliharaan dan menghentikan instance di akhir jendela pemeliharaan jika tidak ada periode berjalan lainnya yang menentukan bahwa instance harus berjalan, dan jika acara pemeliharaan selesai.

  Jika acara pemeliharaan tidak selesai pada akhir jendela pemeliharaan, instance akan berjalan hingga interval penjadwalan setelah acara pemeliharaan selesai.

> **Note**
>
> Penjadwal tidak memvalidasi bahwa sumber daya dimulai atau dihentikan. Itu membuat panggilan API dan melanjutkan. Jika panggilan API gagal, ia mencatat kesalahan untuk penyelidikan.

Pertimbangan keamanan aplikasi

Keamanan aplikasi termasuk mempertimbangkan izin apa yang perlu dijalankan aplikasi, aturan firewall apa, peran IAM apa yang harus diaktifkan untuk akses ke aplikasi.

Untuk lebih memahami AWS keamanan umum, lihat Praktik Terbaik untuk Keamanan, Identitas, & Kepatuhan.

Akses untuk manajemen konfigurasi

AWS Managed Services (AMS) berupaya memberi Anda infrastruktur bebas sakit kepala sehingga Anda tidak perlu khawatir tentang masalah keamanan, masalah penambalan, masalah pencadangan, dll. Untuk melakukan itu, AMS merekomendasikan peran IAM minimal yang memungkinkan hanya grup tertentu atau server master, jika menggunakan alat penerapan aplikasi, akses ke instance yang menjalankan aplikasi Anda.

Aturan firewall akses aplikasi

Sama seperti sistem operasi (OS), semua akses aplikasi harus diatur menggunakan grup Active Directory (AD). Menggunakan Amazon Relational Database Service (Amazon RDS) sebagai contoh, Anda harus memecahkan mirror (replikasi) untuk menambahkan pengguna baru. Pendekatan terbaik adalah membuat grup di AD dan menambahkannya pada waktu pembuatan database. Memiliki grup di AMS AD Anda berarti Anda dapat membuat CTs untuk akses aplikasi. Untuk informasi tentang strategi pengelompokan resmi AD, lihat Menggunakan Strategi Bersarang Grup — Praktik Terbaik AD untuk Strategi Grup.

Untuk mempelajari lebih lanjut tentang pohon domain dan parent/child domain, lihat Cara Kerja Domain dan Hutan.

Aturan berikut menggambarkan solusi yang sesuai untuk kepercayaan hutan multi-domain dengan pengguna yang berada di domain anak.

Contoh Windows

Ini adalah aturan yang harus dikonfigurasi untuk pengontrol domain induk dan anak Windows Anda.

Pengontrol Domain Induk, Windows

DARI: Pengontrol domain induk KE: tumpukan Windows dan subnet layanan bersama

Port Sumber	Pelabuhan Tujuan	Protokol
88	49152 - 65535	TCP
389	49152 - 65535	UDP

DARI: Stack subnet, termasuk layanan bersama KE: Pengontrol domain root hutan Windows

Port Sumber	Pelabuhan Tujuan	Protokol
49152 - 65535	88	TCP
49152 - 65535	389	UDP

Pengontrol Domain Anak, Windows

DARI: Pengontrol domain anak KE: Pengontrol domain AWS Windows

Port Sumber	Pelabuhan Tujuan	Protokol
49152 - 65535	53	TCP
49152 - 65535	88	TCP
49152 - 65535	389	UDP

DARI: Pengontrol domain anak KE: Tumpukan Windows dan subnet layanan bersama

Port Sumber	Pelabuhan Tujuan	Protokol
88	49152 - 65535	TCP
135	49152 - 65535	TCP

Port Sumber	Pelabuhan Tujuan	Protokol
389	49152 - 65535	TCP
389	49152 - 65535	UDP
445	49152 - 65535	TCP
49152 - 65535	49152 - 65535	TCP

DARI: Stack subnet, termasuk layanan bersama KE: Pengontrol domain anak Windows

Port Sumber	Pelabuhan Tujuan	Protokol
49152 - 65535	88	TCP
49152 - 65535	135	TCP
49152 - 65535	389	TCP
49152 - 65535	389	UDP
49152 - 65535	445	TCP
49152 - 65535	49152 - 65535	TCP

Instans Linux

Ini adalah aturan untuk mengkonfigurasi untuk pengontrol domain induk dan anak Linux Anda.

Semua pengujian dilakukan menggunakan Amazon Linux. Sementara rentang port dinamis untuk Windows adalah 49152 hingga 65535, banyak kernel Linux menggunakan rentang port 32768 hingga 61000. Jalankan perintah di bawah ini untuk melihat rentang port IP.

```
cat /proc/sys/net/ipv4/ip_local_port_range
```

Pengontrol Domain Induk, Linux

DARI: Pengontrol domain induk KE: Tumpukan Linux dan subnet layanan bersama

Port Sumber	Pelabuhan Tujuan	Protokol
389	32768 - 61000	UDP
88	32768 - 61000	TCP

DARI: Stack subnet, termasuk layanan bersama KE: Pengontrol domain root hutan Linux

Port Sumber	Pelabuhan Tujuan	Protokol
32768 - 61000	88	TCP
32768 - 61000	389	UDP

Pengontrol Domain Anak, Linux

DARI: Pengontrol domain anak KE: Pengontrol domain AWS Linux

Port Sumber	Pelabuhan Tujuan	Protokol
49152 - 65535	53	TCP
49152 - 65535	88	TCP
389	49152 - 65535	UDP
49152 - 65535	389	UDP

DARI: Pengontrol domain anak KE: Tumpukan Linux dan subnet layanan bersama

Port Sumber	Pelabuhan Tujuan	Protokol
88	32768 - 61000	TCP
389	32768 - 61000	UDP

DARI: Tumpuk subnet, termasuk layanan bersama KE: Pengontrol domain anak Linux

Port Sumber	Pelabuhan Tujuan	Protokol
32768 - 61000	88	TCP
32768 - 61000	389	UDP

Manajemen lalu lintas jalan keluar AMS

Secara default, rute dengan CIDR tujuan 0.0.0.0/0 untuk subnet pribadi AMS dan aplikasi pelanggan memiliki gateway terjemahan alamat jaringan (NAT) sebagai target. Layanan AMS, TrendMicro dan patching, adalah komponen yang harus memiliki akses keluar ke Internet sehingga AMS dapat menyediakan layanannya, dan TrendMicro dan sistem operasi dapat memperoleh pembaruan.

AMS mendukung pengalihan lalu lintas keluar ke internet melalui perangkat jalan keluar yang dikelola pelanggan selama:

- Ini bertindak sebagai proxy implisit (misalnya, transparan).

  and

- Ini memungkinkan dependensi AMS HTTP dan HTTPS (tercantum di bagian ini) untuk memungkinkan penambalan dan pemeliharaan infrastruktur terkelola AMS yang sedang berlangsung.

Beberapa contohnya adalah:

- Gateway transit (TGW) memiliki rute default yang menunjuk ke firewall lokal yang dikelola pelanggan melalui koneksi AWS Direct Connect di akun Jaringan Zona Pendaratan Multi-Akun.
- TGW memiliki rute default yang menunjuk ke titik akhir AWS di VPC keluar Zona Pendaratan Multi-Akun yang memanfaatkan AWS PrivateLink, menunjuk ke proxy yang dikelola pelanggan di akun AWS lain.
- TGW memiliki rute default yang menunjuk ke firewall yang dikelola pelanggan di akun AWS lain, dengan koneksi site-to-site VPN sebagai lampiran ke Multi-Account Landing Zone TGW.

AMS telah mengidentifikasi dependensi HTTP dan HTTPS AMS yang sesuai, dan mengembangkan serta menyempurnakan dependensi ini secara berkelanjutan. Lihat egressMgmt.zip. Seiring dengan file JSON, ZIP berisi README.

> **Note**
>
> - Informasi ini tidak komprehensif - beberapa situs eksternal yang diperlukan tidak tercantum di sini.
> - Jangan gunakan daftar ini di bawah daftar penolakan atau strategi pemblokiran.
> - Daftar ini dimaksudkan sebagai titik awal untuk set aturan penyaringan jalan keluar, dengan harapan bahwa alat pelaporan akan digunakan untuk menentukan dengan tepat di mana lalu lintas aktual menyimpang dari daftar.

Untuk meminta informasi tentang memfilter lalu lintas jalan keluar, kirim email ke CSDM Anda: ams-csdm@amazon.com.

Grup keamanan

Di AWS VPCs, AWS Security Groups bertindak sebagai firewall virtual, mengontrol lalu lintas untuk satu atau beberapa tumpukan (instance atau satu set instance). Ketika tumpukan diluncurkan, itu terkait dengan satu atau beberapa grup keamanan, yang menentukan lalu lintas apa yang diizinkan untuk mencapainya:

- Untuk tumpukan di subnet publik Anda, grup keamanan default menerima lalu lintas dari HTTP (80) dan HTTPS (443) dari semua lokasi (internet). Tumpukan juga menerima lalu lintas SSH dan RDP internal dari jaringan perusahaan Anda, dan benteng AWS. Tumpukan tersebut kemudian dapat keluar melalui port apa pun ke Internet. Mereka juga dapat keluar ke subnet pribadi Anda dan tumpukan lain di subnet publik Anda.
- Tumpukan di subnet pribadi Anda dapat keluar ke tumpukan lain di subnet pribadi Anda, dan instance dalam tumpukan dapat sepenuhnya berkomunikasi melalui protokol apa pun satu sama lain.

> **Important**
>
> Grup keamanan default untuk tumpukan pada subnet pribadi memungkinkan semua tumpukan di subnet pribadi Anda untuk berkomunikasi dengan tumpukan lain di subnet pribadi tersebut. Jika Anda ingin membatasi komunikasi antar tumpukan dalam subnet pribadi, Anda harus membuat grup keamanan baru yang menjelaskan pembatasan tersebut. Misalnya, jika Anda ingin membatasi komunikasi ke server database sehingga tumpukan di subnet pribadi itu hanya dapat berkomunikasi dari server aplikasi tertentu melalui port tertentu, mintalah grup keamanan khusus. Cara melakukannya dijelaskan di bagian ini.

Grup Keamanan Default

MALZ

Tabel berikut menjelaskan pengaturan grup keamanan masuk (SG) default untuk tumpukan Anda. SG diberi nama "SentinelDefaultSecurityGroupPrivateOnly-VPC-ID” yang merupakan ID VPC di *ID* akun landing zone multi-akun AMS Anda. Semua lalu lintas diizinkan keluar ke "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly" melalui grup keamanan ini (semua lalu lintas lokal dalam subnet tumpukan diizinkan).

Semua lalu lintas diizinkan keluar ke 0.0.0.0/0 oleh grup keamanan kedua "". SentinelDefaultSecurityGroupPrivateOnly

> **Tip**
>
> Jika Anda memilih grup keamanan untuk jenis perubahan AMS, seperti EC2 membuat, atau OpenSearch membuat domain, Anda akan menggunakan salah satu grup keamanan default yang dijelaskan di sini, atau grup keamanan yang Anda buat. Anda dapat menemukan daftar grup keamanan, per VPC, di EC2 konsol AWS atau konsol VPC.

Ada grup keamanan default tambahan yang digunakan untuk keperluan AMS internal.

Grup keamanan default AMS (lalu lintas masuk)

Jenis	Protokol	Rentang port	Sumber
Semua Lalu lintas	Semua	Semua	SentinelDefaultSecurityGroupPrivateOnly (membatasi lalu lintas keluar ke anggota grup keamanan yang sama)
Semua Lalu lintas	Semua	Semua	SentinelDefaultSecurityGroupPrivateOnlyEgressAll (tidak membatasi lalu lintas keluar)
HTTP, HTTPS, SSH, RDP	TCP	80/443 (Sumber 0.0.0.0/0) Akses SSH dan RDP diizinkan dari benteng	SentinelDefaultSecurityGroupPublic (tidak membatasi lalu lintas keluar)
Benteng MALZ:			
SSH	TCP	22	SharedServices VPC CIDR dan DMZ VPC CIDR, ditambah pelanggan yang disediakan secara on-prem CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
Benteng SALZ:			
SSH	TCP	22	mc-initial-garden- LinuxBastion SG
SSH	TCP	22	mc-initial-garden- LinuxBastion DMZSG
RDP	TCP	3389	mc-initial-garden- WindowsBastion SG
RDP	TCP	3389	mc-initial-garden- WindowsBastion DMZSG

SALZ

Tabel berikut menjelaskan pengaturan grup keamanan masuk (SG) default untuk tumpukan Anda. SG diberi nama "mc-initial-garden- SentinelDefaultSecurityGroupPrivateOnly -*ID*" di mana *ID* adalah pengenal unik. Semua lalu lintas diizinkan keluar ke "mc-initial-garden-SentinelDefaultSecurityGroupPrivateOnly" melalui grup keamanan ini (semua lalu lintas lokal dalam subnet tumpukan diizinkan).

Semua lalu lintas diizinkan keluar ke 0.0.0.0/0 oleh grup keamanan kedua "- -". mc-initial-garden SentinelDefaultSecurityGroupPrivateOnlyEgressAll *ID*

> **Tip**
>
> Jika Anda memilih grup keamanan untuk jenis perubahan AMS, seperti EC2 membuat, atau OpenSearch membuat domain, Anda akan menggunakan salah satu grup keamanan default yang dijelaskan di sini, atau grup keamanan yang Anda buat. Anda dapat menemukan daftar grup keamanan, per VPC, di EC2 konsol AWS atau konsol VPC.

Ada grup keamanan default tambahan yang digunakan untuk keperluan AMS internal.

Grup keamanan default AMS (lalu lintas masuk)

Jenis	Protokol	Rentang port	Sumber
Semua Lalu lintas	Semua	Semua	SentinelDefaultSecurityGroupPrivateOnly (membatasi lalu lintas keluar ke anggota grup keamanan yang sama)
Semua Lalu lintas	Semua	Semua	SentinelDefaultSecurityGroupPrivateOnlyEgressAll (tidak membatasi lalu lintas keluar)
HTTP, HTTPS, SSH, RDP	TCP	80/443 (Sumber 0.0.0.0/0) Akses SSH dan RDP diizinkan dari benteng	SentinelDefaultSecurityGroupPublic (tidak membatasi lalu lintas keluar)

Jenis	Protokol	Rentang port	Sumber
Benteng MALZ:			
SSH	TCP	22	SharedServices VPC CIDR dan DMZ VPC CIDR, ditambah pelanggan yang disediakan secara on-prem CIDRs
SSH	TCP	22	
RDP	TCP	3389	
RDP	TCP	3389	
Benteng SALZ:			
SSH	TCP	22	mc-initial-garden- LinuxBastion SG
SSH	TCP	22	mc-initial-garden- LinuxBastion DMZSG
RDP	TCP	3389	mc-initial-garden- WindowsBastion SG
RDP	TCP	3389	mc-initial-garden- WindowsBastion DMZSG

Membuat, Mengubah, atau Menghapus Grup Keamanan

Anda dapat meminta grup keamanan khusus. Jika grup keamanan default tidak memenuhi kebutuhan aplikasi atau organisasi Anda, Anda dapat memodifikasi atau membuat grup keamanan baru. Permintaan semacam itu akan dianggap memerlukan persetujuan dan akan ditinjau oleh tim operasi AMS.

Untuk membuat grup keamanan di luar tumpukan dan VPCs, kirimkan RFC menggunakan jenis `Deployment | Advanced stack components | Security group | Create (review required)` perubahan (ct-1oxx2g2d7hc90).

Untuk modifikasi grup keamanan Active Directory (AD), gunakan jenis perubahan berikut:

- Untuk menambahkan pengguna: Kirim RFC menggunakan Manajemen | Directory Service | Pengguna dan grup | Tambahkan pengguna ke grup [ct-24pi85mjtza8k]
- Untuk menghapus pengguna: Kirim RFC menggunakan Manajemen | Directory Service | Pengguna dan grup | Hapus pengguna dari grup [ct-2019s9y3nfml4]

> **Note**
>
> Saat menggunakan “review required” CTs, AMS merekomendasikan agar Anda menggunakan opsi Penjadwalan ASAP (pilih ASAP di konsol, biarkan waktu mulai dan berakhir kosong di API/CLI) karena ini CTs memerlukan operator AMS untuk memeriksa RFC, dan mungkin berkomunikasi dengan Anda sebelum dapat disetujui dan dijalankan. Jika Anda menjadwalkan ini RFCs, pastikan untuk mengizinkan setidaknya 24 jam. Jika persetujuan tidak terjadi sebelum waktu mulai yang dijadwalkan, RFC ditolak secara otomatis.

Temukan Grup Keamanan

Untuk menemukan grup keamanan yang dilampirkan ke tumpukan atau instance, gunakan EC2 konsol. Setelah menemukan tumpukan atau instance, Anda dapat melihat semua grup keamanan yang melekat padanya.

Untuk cara menemukan grup keamanan di baris perintah dan memfilter output, lihat `describe-security-groups`.

Lampiran: Kuesioner orientasi aplikasi

Gunakan kuesioner ini untuk menjelaskan elemen dan struktur penyebaran Anda sehingga AMS dapat menentukan komponen infrastruktur apa yang dibutuhkan. Persyaratan orientasi untuk aplikasi Line-of-Business (LoB) berbeda secara signifikan dari aplikasi produk, jadi kuesioner ini dirancang untuk mengatasi keduanya.

Topik

Ringkasan penyebaran

Deskripsi deployment. Misalnya:

- Akun ini untuk penerapan aplikasi Line-of-Business (LoB) (sebagai lawan dari penerapan aplikasi produk).
- Penyebaran melibatkan ARP berskala otomatis (proxy terbalik yang diautentikasi) dalam subnet akun. public/DMZ
- Server web dan aplikasi akan digunakan dalam subnet pribadi akun.
- Instans Amazon RDS (Amazon Relational Database Service) juga akan digunakan dalam subnet pribadi akun.
- Server (ARP, web, aplikasi, database, load balancer, dan sebagainya) dipisahkan menjadi kelompok-kelompok keamanan yang berbeda.
- Akun tersebut memerlukan desain HA (ketersediaan tinggi) yang tersebar di Availability Zones (AZs), yaitu Multi-AZ.

Komponen penyebaran infrastruktur

Apa saja komponen berbeda yang perlu dikonfigurasi untuk mendukung aplikasi Anda?

- Wilayah: Apa AWS Region atau Wilayah yang dibutuhkan?
- Ketersediaan Tinggi (HA): Zona Ketersediaan Apa yang akan digunakan?
- Virtual Private Cloud (VPC): Apa itu blok CIDR untuk VPC?
- Instance server apa yang dibutuhkan?
  - Authenticated Reverse Proxy (ARP): OS, AMI, tipe instans, subnet ID, grup keamanan, port ingress?
  - Server Alat Penyebaran Aplikasi: OS, AMI, tipe instans, ID subnet, grup keamanan, port masuk (Chef, Puppet) atau port keluar (Ansible, Saltstack) port?
  - Amazon RDS dengan MySQL: versi DB, Jenis Penggunaan, kelas instans, ID subnet, grup keamanan, ID instans DB, ukuran penyimpanan, Multi-AZ, tipe Auth, enkripsi?
  - Penyimpanan: Apakah aplikasi Anda tanpa kewarganegaraan? Apakah Anda memerlukan ember S3? Apakah Anda memerlukan penyimpanan persisten? Apakah Anda memerlukan enkripsi data saat istirahat pada volume EBS Anda? Apakah Anda memerlukan enkripsi DB?
  - Titik akhir server eksternal (ke Managed Services VPC): SMTP? LDAP?
  - Persyaratan jaringan: Pemfilteran jaringan (berdasarkan grup keamanan?)? Inspeksi lalu lintas web (inbound? keluar?)?
- Tag: Tag apa yang harus digunakan untuk mengelompokkan sumber daya ke dalam koleksi logis? Misalnya, semua sumber daya untuk tumpukan aplikasi. Pilih tag untuk kasus penggunaan Anda; misalnya, `backup=true` untuk mengaktifkan cadangan. Selain itu, Anda harus menggunakan tag `name=value` agar setiap EC2 instance yang Anda buat untuk menampilkan nama di konsol.
- Grup keamanan:
  - Kelompok keamanan apa yang dibutuhkan?
  - Aturan masuknya grup keamanan?
  - Aturan keluar kelompok keamanan?

Platform hosting aplikasi

Untuk platform hosting aplikasi Anda, pertimbangkan persyaratan yang mungkin berikut:

- Database dienkripsi?
- Kunci enkripsi dikelola oleh siapa?
- Semua data dalam perjalanan dan istirahat dienkripsi?
- Semua akses pengguna ke sistem melalui HTTPS?

- Semua system-to-system interaksi disetujui oleh tim operasi keamanan Anda?

Model penyebaran aplikasi

Pertimbangan tentang bagaimana Anda merencanakan penerapan aplikasi Anda. Lihat Apa model operasi saya?

- Otomatis atau manual? Tidak ada otomatisasi penerapan berarti tidak ada Skala Otomatis. Jika Anda meminta akses dan masuk dan memperbarui aplikasi Anda secara manual, dan pembaruan Anda gagal. AMS mengharapkan Anda untuk mengembalikan pembaruan Anda atau memberi tahu kami melalui permintaan layanan sehingga kami dapat membantu Anda.
- Jika otomatis, apa kerangka kerjanya? Skrip? Berbasis agen ()? puppet/chef)? Agentless (SALT/ Ansible CodeDeploy? Perkakas penerapan berbasis agen dan tanpa agen memerlukan instance terpisah untuk dibuat dan digunakan sebagai server master untuk perkakas. AMS mengharapkan Anda untuk mengetahui semua elemen yang diperlukan untuk alat penerapan aplikasi yang berhasil; namun, kami dengan senang hati membantu dengan pertanyaan infrastruktur terkait.
- Apakah Line-of-Business aplikasi Anda (aplikasi yang Anda gunakan untuk membuat dan mengelola aplikasi Anda) memerlukan penambalan?

Dependensi aplikasi

Apakah Anda memerlukan instance untuk aplikasi Line-of-Business (LoB)? Untuk aplikasi produk?

Apa yang dibutuhkan aplikasi Produk Anda agar berfungsi dengan baik?

- Dependensi tingkat jaringan: Misalnya, Direct Connect
- Package dependencies: Sebagai contoh, `pip`
- Aplikasi yang bergantung pada aplikasi ini: Misalnya, MySql
- Dependensi firewall?

Apa yang dibutuhkan aplikasi LoB Anda agar berfungsi dengan baik?

- Dependensi tingkat jaringan: Misalnya, Direct Connect
- Package dependencies: Misalnya, Firefox Saucy
- Aplikasi yang bergantung pada aplikasi ini: Misalnya, MySql

- Dependensi firewall?

Sertifikat SSL untuk aplikasi produk

Sertifikat SSL apa yang dibutuhkan server Anda sehingga aplikasi Anda (LoB dan produk) dapat mencapai semua yang mereka butuhkan untuk dijalankan dan dapat diakses?

- Grup Auto Scaling?
- Database (Amazon RDS)?
- Load Balancer?
- Server alat penyebaran?
- Firewall aplikasi web (AWS WAF)?
- Contoh lainnya?

Sebagai contoh, untuk setiap contoh yang tercantum di atas, Anda mungkin memerlukan sertifikat berikut:

WAF (sertifikat 1) -> ELB-ext (sertifikat 2) -> ARP (sertifikat 3) -> ELB-int (sertifikat 4) -> Situs web (sertifikat 5) -> Elb-int (sertifikat 6) -> Layanan web (sertifikat 7).

Riwayat dokumen

Tabel berikut menjelaskan dokumentasi untuk rilis AMS ini.

- Versi API: 2019-05-21
- Pembaruan dokumentasi terbaru: 16 Februari 2023

Ubah	Deskripsi	Tautan
Tautan TOC dihapus	Tautan AWS Glosarium TOC dihapus.	Agustus 08, 2025
Konten yang diperbarui: Memigrasi Beban Kerja: Validasi pra-konsumsi Windows	Bagian yang diperbarui untuk menyertakan langkah-langkah terperinci untuk menggunak an skrip pra WIGs validator untuk memvalida si bahwa instance Windows Anda siap untuk dimasukkan ke dalam akun AMS Anda;	Memigrasi beban kerja: Validasi pra-konsumsi Windows
Konten yang diperbarui, konfigurasi DMS	catatan penting tentang peran yang diperlukan, dms-vpc-role.	1: grup subnet AWS DMS replikasi : Buat
Konten yang diperbarui, sumber daya yang didukung CFN Ingest	Ditambahkan OpenSearch.	Sumber Daya yang Didukung
Konten yang diperbarui, Memigrasi beban kerja	Instruksi yang diperbarui untuk validasi pra-konsumsi.	Memigrasi beban kerja: Validasi pra-konsumsi Windows
Konten yang diperbarui, CFN Ingest.	Menghapus “sumber daya yang didukung” terbatas dari konten konsumsi CFN.	CloudForm ation Ingest Stack: Sumber

Ubah	Deskripsi	Tautan
		daya yang didukung
Diperbarui versi Windows yang didukung	Menambahkan dukungan untuk Windows Server 2022.	Gambar Mesin AMS Amazon (AMIs), Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows, dan Memigrasi beban kerja: Validasi pra-konsumsi Windows
Konten yang diperbarui, Penjadwal Sumber Daya.	Instruksi yang diperbarui untuk menggunakan CT penyebaran khusus, ct-0ywnhc8e5k9z5, berlaku untuk SALZ dan MALZ.	Penjadwal Sumber Daya AMS mulai cepat
Konten yang diperbarui, Workload Ingest.	Diperbarui versi SUSE Linux yang didukung.	Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows
Konten yang diperbarui, Database Migration Service.	Ditambahkan ke prasyarat dan membuat beberapa perubahan untuk kegunaan dan kegunaan.	AWS Database Migration Service (AWS DMS)

Ubah	Deskripsi	Tautan
Konten yang diperbarui, Workload Ingest.	Zip Validasi Pra-wig Linux telah diperbarui.	Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows
Konten yang diperbarui.	Memperbarui zip validasi pra-wigs untuk Linux. Juga, menambahkan Windows Server 2008 R2 sebagai sistem operasi yang didukung.	Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows
Konten baru	Mulai Cepat dan Tutorial telah dipindahkan ke sini dari Panduan Manajemen Perubahan Lanjutan AMS yang sudah pensiun.	Mulai cepat, Tutorial.
Konten diperbarui	Deployment | Komponen tumpukan lanjutan | Database Migration Service (DMS) | Mulai tugas replikasi (ct-1yq7hhqse71yg) Diperbarui untuk menunjukkan DocumentNamedan Wilayah adalah parameter yang diperlukan; sebelumnya, mereka salah terdaftar sebagai opsional.	Database Migration Service (DMS) | Mulai Tugas Replikasi
Konten diperbarui	CloudFormation Tertelan Diperbarui untuk menunjukkan dua sumber daya baru yang didukung, AWS::Route53Resolver::ResolverRuleAssociation dan AWS::Route53Resolver::ResolverRule.	Sumber Daya yang Didukung

Ubah	Deskripsi	Tautan
Konten diperbarui	Memigrasi beban kerja: Validasi pra-konsumsi Windows	Informasi Sysprep diperbarui dengan lebih spesifik. Memigrasi beban kerja: Validasi pra-konsumsi Windows
Konten diperbarui	Manajemen | Tumpukan khusus | Tumpukan dari CloudFormation Template | Menyetuju i Changeset dan Pembaruan (ct-1404e 21baa2ox) Deskripsi panduan CT untuk ChangeSet Nameparameter telah diperbarui dengan informasi tambahan.	Tumpukan dari CloudForm ation Template | Menyetujui Changeset dan Update
	Ubuntu 18.04 dan Oracle Linux 8.3 tersedia	Migrasi Beban Kerja: Prasyarat untuk Linux dan Windows
Konten baru:	Penerapan IAM melalui CFN Ingest dan Pembaruan Stack. CTs	Februari 10, 2022

Ubah	Deskripsi	Tautan
Tugas replikasi Database Migration Service (DMS)	Ubah jenis diperbarui sehingga ekspresi reguler mengizinkan tugas ARNs yang berisi tanda hubung. Mulai AWS DMS tugas replikasidan Database Migration Service (DMS) | Hentikan Tugas Replikasi.	Januari 13, 2022
Validasi pra-konsumsi WIGS Linux	File zip telah diperbarui. Memigrasi beban kerja: Validasi pra-konsumsi Linux.	Januari 13, 2022
Tautan tetap	Database (DB) Impor ke AMS SQL RDS -> Pengaturan bagian memiliki beberapa tautan buruk.	Januari 13, 2022

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.