

Panduan Implementasi

Penjadwal Instans di AWS



Penjadwal Instans di AWS: Panduan Implementasi

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Ikhtisar solusi	1
Fitur dan manfaat	2
Kasus penggunaan	3
Konsep dan definisi	3
Biaya	4
Faktor penskalaan biaya	4
Menghitung target penjadwalan	4
Strategi pengoptimalan biaya	5
Contoh harga referensi (bulanan)	5
Estimasi biaya untuk penerapan Anda	8
Kuota	8
Batasan penskalaan	8
Pertimbangan Tambahan	9
AWS Service Quotas	10
Wilayah AWS yang Didukung	10
Penjadwalan instans lintas akun menggunakan akun atau IDs AWS Organization ID	11
Mengaktifkan penjadwalan lintas akun menggunakan Akun IDs	11
Mengaktifkan penjadwalan lintas akun menggunakan AWS Organization ID	11
Mengelola Akun IDs dengan AWS Systems Manager Parameter Store	12
Layanan yang didukung untuk penjadwalan	12
Perilaku shutdown instance	13
Amazon EC2	13
Amazon RDS, Amazon Neptune, dan Amazon DocumentDB	13
Jendela pemeliharaan Amazon RDS	13
Grup Amazon EC2 Auto Scaling	14
Arsitektur	15
Diagram arsitektur	15
Pertimbangan desain AWS Well-Architected	17
Keunggulan operasional	17
Keamanan	18
Keandalan	18
Efisiensi kinerja	18
Optimalisasi biaya	19
Keberlanjutan	19

Tabel konfigurasi penjadwal	19
Penjadwal CLI	20
Layanan AWS yang digunakan dalam solusi ini	20
Keamanan	22
AWS KMS	22
Amazon IAM	22
Volume EC2 EBS Terenkripsi	23
License Manager EC2	24
Memulai	26
Ikhtisar proses penyebaran	26
CloudFormation Templat AWS	27
Langkah 1: Luncurkan stack hub scheduler instance	27
Langkah 2 (Opsional): Luncurkan tumpukan jarak jauh di akun sekunder	34
Konfigurasi solusinya	37
Panduan operator	38
Konfigurasi jadwal	38
Menggunakan Infrastruktur sebagai Kode (disarankan)	38
Menggunakan Amazon DynamoDB Console dan Instance Scheduler di AWS CLI	38
Tandai contoh untuk penjadwalan	39
Mengatur nilai tag	39
Instans EC2 dengan volume EBS terenkripsi	40
Instans EC2 dikelola di License Manager	40
Referensi jadwal	40
Periode	40
Zona waktu	41
Menghentikan bidang instance baru	41
Bidang hibernasi	41
Bidang yang ditegakkan	41
Pertahankan bidang yang sedang berjalan	42
Bidang jendela pemeliharaan Systems Manager (hanya berlaku untuk instans EC2)	42
Tipe instans	43
Definisi jadwal	43
Referensi periode	46
Mulai dan hentikan waktu	46
Hari dalam seminggu	48
Hari dalam sebulan	48

Bulan	48
Definisi periode	48
Jadwal sampel	51
Standar 9-5 jam kerja	52
Hentikan instance setelah jam 5 sore	54
Hentikan contoh selama akhir pekan	56
Sumber daya solusi	59
Penjadwal CLI	60
Prasyarat	60
Kredensial	60
Instal CLI Penjadwal	61
Struktur perintah	62
Argumen umum	62
Perintah yang tersedia	63
buat-periode	63
buat-jadwal	66
hapus-periode	68
hapus-jadwal	69
jelaskan-periode	69
jelaskan-jadwal	71
describe-schedule-usage	72
periode pembaruan	73
jadwal pembaruan-	74
bantuan	74
Perbarui pengaturan konfigurasi global	75
Mengelola jadwal menggunakan Infrastructure as Code (IaC)	76
Menangani Kesalahan Kapasitas Tidak Cukup EC2	78
Konfigurasi	78
Cara kerjanya	78
Persyaratan dan pembatasan	79
Contoh	79
Penjadwalan Grup Penskalaan Otomatis EC2	79
Ikhtisar penjadwalan ASG	80
Definisi Running/Stopped untuk ASGs	80
Perilaku ASG Start/Stop	80
Pantau solusinya	81

Pencatatan dan pemberitahuan	81
CloudWatch Kueri Wawasan Log	81
Dasbor wawasan operasional	82
Monitoring EventBridge Acara	84
Pemecahan masalah	87
Resolusi masalah yang diketahui	87
Masalah: Instans tidak dijadwalkan di akun jarak jauh (v1.4-v3.0)	87
Resolusi	87
Masalah: Contoh tidak dijadwalkan (v3.1 +)	88
Resolusi	88
Masalah: Instans EC2 terenkripsi tidak dimulai	88
Resolusi	88
Masalah: Instans RDS tidak berhenti saat Buat Snapshot RDS Diaktifkan	88
Resolusi	88
Hubungi AWS Support	89
Buat kasus	89
Bagaimana kami bisa membantu?	89
Informasi tambahan	89
Bantu kami menyelesaikan kasus Anda lebih cepat	89
Selesaikan sekarang atau hubungi kami	90
Perbarui solusinya	91
Melanggar Perubahan dalam Versi Tertentu	92
v1.5.0	92
v3.0.0	92
v3.1.0	93
Copot pemasangan solusinya	95
Menggunakan Konsol Manajemen AWS	95
Menggunakan AWS Command Line Interface	95
Panduan developer	97
Kode sumber	97
Referensi	98
Pengumpulan data	98
Sumber daya terkait	98
Kontributor	99
Revisi	101
Pemberitahuan	102

..... **ciii**

Mengotomatiskan memulai dan menghentikan instans AWS

Solusi Penjadwal Instans di AWS mengotomatiskan memulai dan menghentikan berbagai layanan AWS termasuk instans [Amazon Elastic Compute Cloud](#) (Amazon EC2) dan Amazon [Relational Database Service](#) (Amazon RDS).

Solusi ini membantu mengurangi biaya operasional dengan menghentikan sumber daya yang tidak digunakan dan memulai sumber daya ketika kapasitas mereka dibutuhkan. Misalnya, perusahaan dapat menggunakan Penjadwal Instance di AWS untuk menghentikan instans secara otomatis di luar jam kerja setiap hari. Jika Anda membiarkan semua instans Anda berjalan pada pemanfaatan penuh, solusi ini dapat menghasilkan penghematan biaya hingga 70% untuk kasus-kasus yang hanya diperlukan selama jam kerja reguler (pemanfaatan mingguan dikurangi dari 168 jam menjadi 50 jam).

Penjadwal Instans di AWS memanfaatkan tag sumber daya Amazon Web Services (AWS) dan [AWS Lambda](#) untuk menghentikan dan memulai ulang instans secara otomatis di beberapa Wilayah AWS dan akun pada jadwal yang ditentukan pelanggan. Solusi ini juga memungkinkan Anda menggunakan hibernasi untuk instans EC2 yang dihentikan.

Panduan implementasi ini memberikan gambaran umum tentang solusi Penjadwal Instans pada AWS, arsitektur referensi dan komponennya, pertimbangan untuk merencanakan penerapan, dan langkah-langkah konfigurasi untuk menerapkan solusi ke AWS Cloud.

Panduan ini ditujukan untuk arsitek infrastruktur TI, administrator, dan DevOps profesional yang ingin menerapkan Penjadwal Instance di AWS di lingkungan mereka.

Gunakan tabel navigasi ini untuk menemukan jawaban atas pertanyaan-pertanyaan ini dengan cepat:

Jika kau mau.	Baca.
Ketahui biaya untuk menjalankan solusi ini. Perkiraan biaya untuk menjalankan solusi ini di Wilayah AS Timur (Virginia N.) adalah USD \$13,15 per bulan.	Biaya
Pahami pertimbangan keamanan untuk solusi ini.	AWS Well-Architected Keamanan, Keamanan

Jika kau mau.	Baca.
Konfigurasi jadwal.	Tabel konfigurasi penjadwal
Ketahui Wilayah AWS mana yang didukung untuk solusi ini.	Wilayah AWS yang Didukung
Lihat atau unduh CloudFormation templat AWS yang disertakan dalam solusi ini untuk secara otomatis menerapkan sumber daya infrastruktur (“tumpukan”) untuk solusi ini.	CloudFormation Templat AWS
Akses kode sumber dan secara opsional gunakan AWS Cloud Development Kit (AWS CDK) untuk menerapkan solusi.	GitHub repositori

Fitur dan manfaat

Jadwal Instans pada solusi AWS menyediakan fitur-fitur berikut:

Penjadwalan instans lintas akun

Solusi ini mencakup template yang membuat peran [AWS Identity and Access Management \(IAM\)](#) yang diperlukan untuk memulai dan menghentikan instance di akun sekunder. Untuk informasi selengkapnya, lihat bagian [penjadwalan instans lintas akun](#).

Penandaan otomatis

Penjadwal Instance di AWS dapat secara otomatis menambahkan tag ke semua instance yang dimulai atau dihentikan. Solusinya juga mencakup makro yang memungkinkan Anda menambahkan informasi variabel ke tag.

Konfigurasi jadwal atau periode menggunakan Scheduler CLI

Solusi ini mencakup antarmuka baris perintah (CLI) yang menyediakan perintah untuk mengkonfigurasi jadwal dan periode. CLI memungkinkan pelanggan untuk memperkirakan penghematan biaya untuk jadwal tertentu. Untuk informasi lebih lanjut, lihat [CLI Penjadwal](#).

Mengelola jadwal menggunakan Infrastructure as Code (IaC)

Solusi ini menyediakan Sumber Daya CloudFormation Kustom AWS yang dapat Anda gunakan untuk mengelola jadwal menggunakan Infrastructure as Code (IaC). Untuk informasi selengkapnya, lihat [Mengelola Jadwal Menggunakan Infrastruktur sebagai Kode](#).

Integrasi dengan Windows Pemeliharaan Systems Manager

Untuk instans Amazon EC2, Penjadwal Instans di AWS dapat diintegrasikan dengan jendela pemeliharaan [AWS Systems](#) Manager, yang ditentukan di Wilayah yang sama dengan instans tersebut, untuk memulai dan menghentikannya sesuai dengan jendela pemeliharaan.

Kasus penggunaan

Menjalankan instance hanya selama jam kerja

Jika Anda membiarkan semua instans Anda berjalan pada pemanfaatan penuh, solusi ini dapat menghasilkan penghematan biaya hingga 76% untuk kasus-kasus yang hanya diperlukan selama jam kerja reguler (pemanfaatan mingguan dikurangi dari 168 jam menjadi 40 jam). Untuk informasi selengkapnya, lihat [Jadwal sampel](#).

Menghentikan instance setelah jam kerja

Jika Anda ingin memastikan bahwa instance pengembangan dimatikan setelah jam kerja dan sampai diperlukan lagi, Anda dapat menggunakan solusi ini untuk menetapkan periode akhir tanpa periode mulai. Untuk informasi selengkapnya, lihat [Jadwal sampel](#).

Konsep dan definisi

Bagian ini menjelaskan konsep-konsep kunci dan mendefinisikan terminologi khusus untuk solusi ini:

jadwal

Kelompok satu atau lebih periode yang terikat oleh suatu instance.

periode

Periode berjalan ditentukan oleh waktu mulai dan berhenti.

contoh

Sumber daya yang didukung yang dapat dijadwalkan. Misalnya, instans Amazon EC2 atau cluster Amazon RDS Amazon EC2 dan Amazon RDS.

jam kerja reguler

9:00 hingga 17:00 (9 AM - 5 PM) ET pada hari kerja

Untuk referensi umum istilah AWS, lihat [Daftar Istilah AWS](#).

Biaya

Anda bertanggung jawab atas biaya layanan AWS yang digunakan saat menjalankan Penjadwal Instance. Memahami bagaimana skala biaya dengan ukuran penerapan membantu Anda merencanakan dan mengoptimalkan implementasi Anda.

Faktor penskalaan biaya

Skala biaya Penjadwal Instance berdasarkan beberapa faktor:

Jumlah target penjadwalan: Jumlah account-region-service kombinasi unik yang dikelola. Setiap target memerlukan pemanggilan Lambda terpisah per interval penjadwalan.

Sumber daya per target: Jumlah sumber daya (instans EC2, database RDS, dll.) Dalam setiap target memengaruhi waktu eksekusi Lambda dan biaya durasi.

Kompleksitas metrik operasional: Skala biaya CloudWatch metrik opsional dengan jumlah jenis instans unik dan jadwal aktif yang dilacak di seluruh penerapan Anda.

Frekuensi penjadwalan: Solusi berjalan berdasarkan frekuensi yang dikonfigurasi (default: 5 menit). Pemeriksaan yang lebih sering meningkatkan pemanggilan Lambda dari 24 kali sehari (per jam) menjadi 288 kali sehari (interval 5 menit).

Menghitung target penjadwalan

Target penjadwalan adalah kombinasi unik account-region-service yang berisi setidaknya satu instance yang dikelola secara aktif. Beberapa instance dalam account-region-service kombinasi yang sama dihitung sebagai target penjadwalan tunggal.

Contoh perhitungan:

- Akun A, us-east-1, 5 instans EC2 = 1 target penjadwalan
- Akun A, us-east-1, 3 database RDS = 1 target penjadwalan
- Akun A, us-east-1, 2 grup Auto Scaling = 1 target penjadwalan

- Akun A, us-west-2, 2 instans EC2 = 1 target penjadwalan
- Akun B, us-east-1, 10 instans EC2 = 1 target penjadwalan

Total: 5 target penjadwalan

Ini berarti solusi akan memanggil 5 fungsi Lambda terpisah per interval penjadwalan untuk mengelola semua sumber daya di seluruh kombinasi ini. `account-region-service`

Note

Target dapat berada dalam ruang lingkup untuk penjadwalan tetapi tidak dianggap “aktif” untuk perhitungan biaya sampai setidaknya satu sumber daya ditandai untuk penjadwalan dalam target tersebut.

Untuk pengoptimalan biaya, Penjadwal Instance mengelompokkan semua layanan terkait Amazon RDS ke dalam satu pemanggilan. Oleh karena itu Amazon RDS, [Amazon Aurora](#), [Amazon Neptunus](#), dan [Amazon DocDB](#) penjadwalan semuanya dihitung sebagai satu layanan “RDS” untuk perhitungan biaya.

Strategi pengoptimalan biaya

1. Terapkan di Wilayah dengan harga Lambda yang lebih rendah
2. Gunakan pengaturan memori Lambda 512 MB default kecuali diperlukan untuk meningkatkannya dengan batasan skala target tunggal
3. Minimalkan jumlah jadwal unik dan jenis instance yang digunakan secara aktif
4. Sesuaikan frekuensi penjadwalan berdasarkan kebutuhan Anda
5. Nonaktifkan dasbor metrik operasional jika tidak berencana menggunakannya

Lihat halaman web harga untuk setiap [layanan AWS dalam solusi ini](#).

Sebaiknya buat [anggaran](#) melalui AWS Cost Explorer untuk membantu mengelola biaya. Harga dapat berubah sewaktu-waktu.

Contoh harga referensi (bulanan)

Contoh berikut menunjukkan bagaimana skala biaya di berbagai ukuran penerapan. Gunakan ini sebagai titik referensi untuk memperkirakan biaya penerapan spesifik Anda.

Note

Semua harga referensi adalah estimasi kasar biaya untuk layanan utama yang digunakan oleh solusi.

Penerapan kecil (~ \$9 per bulan)

Contoh ini mewakili pengembangan tipikal atau penyebaran produksi kecil:

- 5 target aktif
- 20 sumber daya terkelola
- 3 jadwal aktif
- 2 jenis contoh
- Interval penjadwalan 5 menit
- Fungsi Lambda 512 MB, runtime rata-rata 5 detik

AWS service	Biaya Bulanan [USD]
AWS Lambda	~ \$2,00
AWS KMS	~ \$1,50
CloudWatch Log	~ \$0,30
CloudWatch Metrik	~\$5,30
Amazon DynamoDB	~\$0,05
Jumlah:	~\$9,15

Penerapan sedang (~ \$161 per bulan)

Contoh ini mewakili penyebaran perusahaan menengah:

- 250 target aktif

- 1000 sumber daya terkelola
- 15 jadwal aktif
- 15 jenis contoh
- Interval penjadwalan 5 menit
- Fungsi Lambda 512 MB, runtime rata-rata 5 detik
- 5 Pemeliharaan EC2 Windows

AWS service	Biaya Bulanan [USD]
AWS Lambda	~ \$95,00
Amazon DynamoDB	~ \$1,00
CloudWatch Log	~ \$10,00
CloudWatch Metrik	~ \$40,00
AWS KMS	~ \$15,00
Jumlah:	~\$161,00

Penerapan besar (~ \$630 per bulan)

Contoh ini mewakili penyebaran perusahaan besar:

- 1000 target aktif
- 5000 sumber daya terkelola
- 500 jadwal aktif
- 50 jenis contoh
- Interval penjadwalan 5 menit
- Fungsi Lambda 512 MB, runtime rata-rata 5 detik
- 100 EC2 Pemeliharaan Windows

AWS service	Biaya Bulanan [USD]
AWS Lambda	~ \$380,00
Amazon DynamoDB	~ \$5,00
CloudWatch Log	~ \$50,00
CloudWatch Metrik	~\$140,00
AWS KMS	~ \$55,00
Jumlah:	~ \$630,00

Estimasi biaya untuk penerapan Anda

Untuk memperkirakan biaya untuk penerapan spesifik Anda:

1. Hitung total sumber daya terkelola Anda (instans EC2, database RDS, dll.)
2. Tentukan jumlah akun dan Wilayah yang akan Anda kelola
3. Pertimbangkan frekuensi penjadwalan yang Anda butuhkan
4. Putuskan apakah Anda memerlukan metrik operasional
5. Gunakan contoh referensi di atas untuk menginterpolasi biaya yang Anda harapkan

Kuota

Batasan penskalaan

Skala Penjadwal Instance pada dua sumbu utama untuk mengelola penerapan perusahaan besar:

Penskalaan vertikal (sumber daya per target)

Penskalaan vertikal dibatasi oleh jumlah sumber daya yang dapat diproses oleh satu fungsi Lambda. Permintaan Penjadwalan secara efisien dalam satu target penjadwalan (kombinasi). `account/region/service`

Instance Scheduler dirancang untuk dapat menangani 1000-an, 100-an EC2s ASGs, dan 100-an RDS dbs/clusters dalam satu [target penjadwalan](#) tetapi mungkin dibatasi oleh latensi lintas wilayah.

Untuk memastikan kinerja yang optimal, kami sarankan untuk memantau waktu eksekusi Lambda Permintaan Penjadwalan ([lihat dasbor wawasan operasional](#)). Sebaiknya pertahankan runtime rata-rata di bawah 90 detik dengan waktu puncak maksimum tidak lebih dari 4 menit.

Penskalaan horisontal (jumlah target)

Penskalaan horizontal dibatasi oleh jumlah [target penjadwalan aktif](#) yang dikelola. Target aktif adalah account/region/service kombinasi dengan setidaknya satu sumber daya yang ditandai secara aktif. Penjadwal Instance dapat digunakan ke lebih banyak akun dan wilayah, tetapi hanya target dengan sumber daya yang ditandai secara aktif yang memengaruhi kinerja.

Dengan kuota konkurensi Lambda default 1000, Anda dapat menjalankan 1000 target aktif secara bersamaan. Lambda mengantri eksekusi tambahan secara otomatis, memungkinkan Anda untuk melampaui batas ini. Kami merekomendasikan untuk menjaga penundaan kumulatif di bawah 3 menit untuk kinerja yang optimal.

Misalnya, dengan runtime rata-rata 15 detik dan batas konkurensi lambda 1000, Anda dapat mengelola hingga 12000 target aktif sambil menjaga penundaan kumulatif di bawah 3 menit ($3 \text{ menit} \div 15 \text{ detik} \times 1000 = 12000 \text{ target}$).

Untuk penerapan yang lebih besar, Anda dapat meminta peningkatan kuota konkurensi Lambda dari AWS Support.

Pertimbangan Tambahan

Tag sumber daya AWS: Sumber daya AWS biasanya memiliki batas 50 tag per sumber daya. Penjadwal Instance memerlukan 6 tag informasi dan kontrol untuk mengoperasikan solusi. Pastikan sumber daya Anda memiliki kapasitas tag yang cukup untuk mengakomodasi tag Penjadwal Instance dan strategi penandaan yang ada.

Batas eksekusi Lambda: Setiap fungsi Lambda Handler Permintaan Penjadwal memiliki batas waktu eksekusi 5 menit.

Penskalaan DynamoDB: Solusinya menggunakan penskalaan sesuai permintaan untuk tabel [Amazon DynamoDB](#), secara otomatis menyesuaikan kapasitas berdasarkan beban kerja Anda.

Batas laju API: Pelambatan API layanan AWS dapat terjadi dengan penerapan yang sangat besar. Solusinya mencakup logika coba lagi untuk menangani pelambatan sementara, tetapi pelambatan yang berlebihan dapat mengurangi batas penskalaan atas solusi.

AWS Service Quotas

Kuota layanan untuk layanan AWS

Service quotas, juga disebut batasan, adalah jumlah maksimum sumber daya layanan atau operasi untuk akun AWS Anda. Pastikan Anda memiliki kuota yang cukup untuk setiap layanan yang diterapkan dalam solusi ini. Untuk informasi lebih lanjut, lihat [Service quotas AWS](#).

CloudFormation Kuota AWS

Akun AWS Anda memiliki CloudFormation kuota AWS yang harus Anda ketahui saat meluncurkan tumpukan dalam solusi ini. Dengan memahami kuota ini, Anda dapat menghindari kesalahan pembatasan yang akan mencegah Anda menerapkan solusi ini dengan sukses. Untuk informasi selengkapnya, lihat [CloudFormation kuota AWS](#) di Panduan CloudFormation Pengguna AWS.

Kuota AWS Lambda

Akun Anda memiliki kuota Eksekusi Serentak AWS Lambda default sebesar 1000. Untuk penerapan yang lebih besar, sebaiknya gunakan Penjadwal Instance ke akun khusus agar tidak bersaing dengan beban kerja lain untuk konkurensi Lambda. Nilai ini dapat disesuaikan. Untuk informasi selengkapnya, lihat panduan [Memulai AWS Lambda](#).

Wilayah AWS yang Didukung

Anda dapat menerapkan Penjadwal Instance di Wilayah AWS mana pun, termasuk Wilayah AWS GovCloud (AS) dan beberapa Wilayah [keikutsertaan \(Wilayah\)](#) yang dinonaktifkan secara default). Setelah menerapkan solusi, Anda dapat mengonfigurasinya untuk menerapkan tindakan mulai atau berhenti yang sesuai ke instans EC2 dan RDS DB yang ditandai di Wilayah mana pun di akun Anda. Jika Anda menggunakan penjadwalan instans lintas akun, solusinya akan menerapkan tindakan ke instance di semua Wilayah yang dikonfigurasi di semua akun.

Important

Penjadwal Instans pada tindakan AWS memengaruhi instans yang diberi tag dengan tepat di semua Wilayah AWS akun Anda, meskipun fungsi Lambda berjalan di satu Wilayah.

Anda dapat menggunakan beberapa penerapan solusi untuk menjadwalkan sejumlah besar instans, atau instance di banyak akun dan Wilayah. Saat Anda menerapkan beberapa penjadwal, gunakan

nama tag yang berbeda untuk setiap tumpukan, dan konfigurasi satu set Wilayah yang tidak tumpang tindih untuk setiap penerapan.

Setiap penerapan memeriksa setiap instance di setiap Wilayah yang dikonfigurasi dalam akun untuk kunci tag yang mengidentifikasi sumber daya yang harus dijadwalkan. Jika Wilayah untuk beberapa penerapan tumpang tindih, setiap instance akan diperiksa oleh beberapa penerapan.

Note

Penjadwal Instans di AWS dapat menargetkan instans dalam wilayah keikutsertaan apa pun untuk penjadwalan, bahkan ketika tumpukan solusi diterapkan di wilayah AWS standar.

Penjadwalan instans lintas akun menggunakan akun atau IDs AWS Organization ID

Solusi ini mencakup template ([instance-scheduler-on-aws-remote.template](#)) yang membuat peran [AWS Identity and Access Management \(IAM\)](#) dan sumber daya lain yang diperlukan untuk memungkinkan solusi memulai penjadwalan di akun sekunder. Anda dapat meninjau dan memodifikasi izin di template jarak jauh sebelum meluncurkan tumpukan.

Mengaktifkan penjadwalan lintas akun menggunakan Akun IDs

Untuk menerapkan jadwal start-stop otomatis ke sumber daya di akun sekunder:

1. Masuk ke [AWS Management Console](#) dan pilih tombol untuk meluncurkan CloudFormation template [instance-scheduler-on-aws](#) di akun utama.
2. Luncurkan template jarak jauh ([instance-scheduler-on-aws-remote](#)) di setiap akun sekunder yang berlaku. Ketika setiap tumpukan jarak jauh diluncurkan, itu membuat peran lintas akun Amazon Resource Name (ARN).
3. Perbarui tumpukan solusi utama dengan ID Akun di IDs parameter Menyediakan ID Organisasi atau Daftar Akun Jarak Jauh untuk memungkinkan solusi melakukan tindakan mulai dan menghentikan pada instance di akun sekunder.

Mengaktifkan penjadwalan lintas akun menggunakan AWS Organization ID

Untuk menerapkan jadwal start-stop otomatis ke sumber daya di akun sekunder:

1. Masuk ke [AWS Management Console](#) dan pilih tombol untuk meluncurkan CloudFormation template [instance-scheduler-on-aws](#) di akun utama.
2. Tetapkan CloudFormation parameter Menggunakan AWS Organizations? sebagai Ya, dan berikan ID organisasi dalam Menyediakan ID Organisasi ATAU Daftar IDs CloudFormation parameter Akun Jarak Jauh.
3. Setelah menerapkan tumpukan di akun utama, luncurkan template jarak jauh (`instance-scheduler-on-aws-remote`) di setiap akun sekunder yang berlaku di Wilayah yang sama dengan solusi di akun utama. Ketika setiap tumpukan jarak jauh berhasil diluncurkan, akun solusi utama akan diperbarui dengan id akun tanpa perubahan lebih lanjut di akun utama.

Mengelola Akun IDs dengan AWS Systems Manager Parameter Store

Gunakan AWS Systems Manager Parameter Store untuk menyimpan akun jarak jauh IDs. Anda dapat menyimpan Akun jarak jauh IDs sebagai parameter daftar di mana setiap item adalah ID akun, atau sebagai parameter string yang berisi daftar akun jarak jauh yang dibatasi koma. IDs Parameter memiliki format `{param:_name_}` di mana nama adalah nama parameter di Parameter Store.

Untuk memanfaatkan fitur ini, Anda harus meluncurkan Penjadwal Instans di tumpukan hub AWS di akun yang sama dengan penyimpanan parameter Anda.

Layanan yang didukung untuk penjadwalan

Penjadwal Instans di AWS saat ini mendukung penjadwalan layanan berikut:

- Amazon EC2
- Grup Amazon EC2 Auto Scaling
- Amazon RDS
- Cluster Amazon Aurora
- Amazon DocumentDB
- Amazon Neptune

Perilaku shutdown instance

Amazon EC2

Solusi ini dirancang untuk menghentikan instans EC2 secara otomatis dan mengasumsikan bahwa perilaku shutdown instance disetel ke Stop, bukan Terminate. Perhatikan bahwa Anda tidak dapat memulai ulang instans Amazon EC2 setelah dihentikan.

Secara default, instans EC2 dikonfigurasi untuk berhenti, bukan menghentikan, saat dimatikan, tetapi Anda dapat [mengubah](#) perilaku ini. Oleh karena itu, pastikan instans yang Anda kontrol menggunakan Penjadwal Instance di AWS dikonfigurasi dengan perilaku Stop shutdown; jika tidak, instans tersebut akan dihentikan.

Amazon RDS, Amazon Neptune, dan Amazon DocumentDB

Solusi ini dirancang untuk secara otomatis menghentikan, bukan menghapus, RDS, Neptune, dan instance DocDB. Anda dapat menggunakan parameter CloudFormation template Create RDS Instance Snapshot AWS untuk membuat snapshot instans RDS DB sebelum solusi menghentikan instans. Snapshot disimpan hingga waktu berikutnya instance dihentikan dan snapshot baru dibuat.

Note

Snapshot tidak tersedia untuk klaster Amazon Aurora. Anda dapat menggunakan parameter template Schedule Aurora Clusters untuk memulai dan menghentikan instans RDS DB yang merupakan bagian dari cluster Aurora atau yang mengelola database Aurora. Anda harus menandai cluster (bukan instance individual) dengan kunci tag yang Anda tentukan selama konfigurasi awal dan nama jadwal sebagai nilai tag untuk menjadwalkan klaster tersebut.

Untuk informasi selengkapnya tentang batasan untuk memulai dan menghentikan instans RDS DB, lihat [Menghentikan instans Amazon RDS DB sementara](#) di Panduan Pengguna Amazon RDS.

Ketika instans RDS DB dihentikan, cache dihapus, yang dapat menyebabkan kinerja lebih lambat saat instance dimulai ulang.

Jendela pemeliharaan Amazon RDS

Setiap instans RDS DB 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. Amazon RDS tidak akan menghentikan instance setelah acara pemeliharaan selesai.

Solusinya memungkinkan Anda untuk menentukan apakah akan menambahkan jendela pemeliharaan yang disukai dari instans RDS DB 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. Untuk informasi selengkapnya tentang jendela pemeliharaan Amazon RDS, lihat [Mempertahankan instans DB](#) di Panduan Pengguna Amazon RDS.

Grup Amazon EC2 Auto Scaling

Kami merancang solusi ini untuk menghentikan grup Amazon EC2 Auto Scaling secara otomatis dengan menggunakan tindakan penskalaan terjadwal. Anda dapat menggunakan solusi untuk mengonfigurasi tindakan penskalaan terjadwal pada grup Auto Scaling (ASG). Ketika ASG dihentikan oleh tindakan penskalaan terjadwal, kapasitas minimum, yang diinginkan, dan maksimum akan diatur 0 hingga ASG dimulai kembali secara otomatis. Ini akan mengembalikan kapasitas minimum, yang diinginkan, dan maksimum ke nilai aslinya.

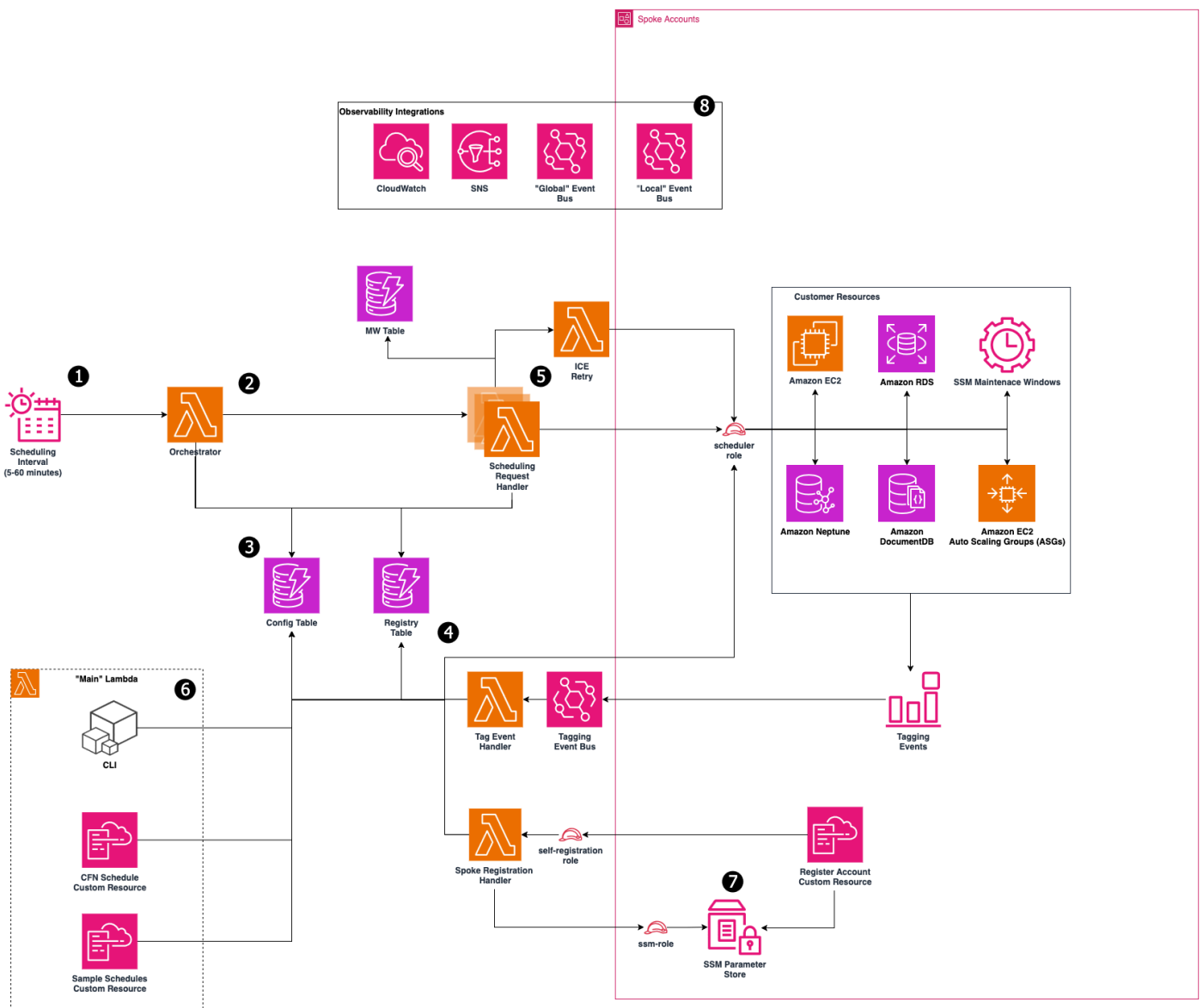
Arsitektur

Bagian ini menyediakan diagram arsitektur implementasi referensi, pertimbangan [desain AWS Well-Architected](#), komponen [keamanan](#), [konfigurasi penjadwal](#), dan [layanan AWS](#) yang digunakan dalam solusi ini.


Diagram arsitektur

Menerapkan solusi ini akan menerapkan komponen berikut di akun AWS Anda.

Penjadwal Instans di AWS Cloud



1. [EventBridge Aturan Amazon](#) memicu fungsi Lambda orkestrasi pada interval yang dapat dikonfigurasi (default: setiap 5 menit).
2. EventBridge [Aturan tersebut memanggil fungsi orkestrasi AWS Lambda yang menanyakan tabel konfigurasi DynamoDB untuk mengidentifikasi target penjadwalan aktif](#). Orkestrator kemudian memanggil fungsi Lambda penjadwalan paralel untuk setiap target aktif.
3. Definisi dan periode jadwal disimpan dalam tabel konfigurasi [Amazon DynamoDB](#). Anda dapat menentukan sejumlah jadwal dan periode dalam tabel ini untuk mengontrol kapan instans Anda mulai dan berhenti.
4. Tabel registri DynamoDB secara otomatis melacak semua sumber daya yang dikelola. Saat sumber daya diberi [tag untuk penjadwalan](#), sumber daya tersebut terdaftar di tabel ini sebagai respons terhadap peristiwa penandaan AWS.
5. Setiap fungsi penjadwalan Lambda menjelaskan sumber daya yang ditandai, mengevaluasi jadwal mereka terhadap waktu saat ini, dan mengeksekusi tindakan mulai atau berhenti yang sesuai.
 - a. Untuk instans EC2, jika operasi awal gagal karena kapasitas yang tidak mencukupi, solusi dapat dikonfigurasi untuk mencoba mengubah ukuran instance menjadi [tipe instans alternatif](#) sebelum mencoba kembali operasi mulai.
6. [Manajemen jadwal tersedia melalui konsol DynamoDB, alat CLI penjadwal, atau sumber daya AWS Custom. CloudFormation](#) Solusi ini digunakan dengan beberapa contoh jadwal yang telah dikonfigurasi sebelumnya.
7. Penerapan lintas akun menggunakan arsitektur hub-spoke di mana akun spoke secara otomatis mendaftar sendiri dengan akun hub. [Tumpukan Spoke harus diterapkan di wilayah yang sama dengan tumpukan hub dan harus disetujui sebelumnya oleh tumpukan hub atau anggota Organisasi AWS yang sama](#).
8. Solusi ini menerbitkan [acara penjadwalan dan pendaftaran](#) ke EventBridge bus di akun hub (acara global) dan akun bicara (acara lokal per wilayah).

 Note

CloudFormation Sumber daya AWS dibuat dari konstruksi [\(AWS CDK\)](#).

Semua fungsi Lambda yang digunakan oleh solusi ini memanfaatkan AWS IAM untuk persyaratan izin sumber daya Anda, dan AWS KMS untuk enkripsi Layanan Pemberitahuan [Sederhana Amazon](#) ([topik Amazon](#) SNS) dan tabel DynamoDB.

Setiap kali solusi melakukan interval penjadwalan, ia memeriksa status saat ini dari setiap instance yang diberi tag dengan tepat terhadap status yang ditargetkan (ditetapkan oleh satu atau beberapa [periode](#) dalam jadwal dalam tag instance) dalam jadwal terkait. Interval jadwal kemudian menerapkan tindakan mulai atau berhenti yang sesuai, seperlunya.

Misalnya, jika fungsi Lambda dipanggil pada hari Jumat pukul 9 pagi (ET) dan mengidentifikasi instans EC2 atau RDS DB yang dihentikan dengan tag `schedule=office-hours`, ia akan memeriksa Amazon DynamoDB untuk detail konfigurasi jadwal jam kantor. Jika jadwal jam kantor berisi periode yang menunjukkan bahwa instance harus berjalan dari Senin hingga Jumat dari jam 9 pagi ET hingga 5 sore ET, fungsi Lambda akan memulai instance itu.

Fungsi Lambda juga merekam informasi tentang sumber daya Anda dan menampilkannya di dasbor [CloudWatch Kustom Amazon opsional](#). Informasi yang direkam mencakup jumlah instance yang ditandai untuk setiap jadwal, ukuran instance tersebut, dan apakah instans tersebut saat ini dalam keadaan berjalan atau berhenti. Untuk informasi selengkapnya tentang dasbor khusus ini, lihat [Dasbor wawasan operasional](#).

Note

Menghentikan instans Amazon EC2 berbeda dengan menghentikan instans Amazon EC2. Secara default, instans Amazon EC2 dikonfigurasi untuk berhenti, bukan menghentikan, saat dimatikan, tetapi Anda dapat mengubah perilaku ini. Sebelum menggunakan solusi ini, verifikasi bahwa instance diatur untuk berhenti atau mengakhiri sebagaimana mestinya.

Pertimbangan desain AWS Well-Architected

Kami merancang solusi ini dengan praktik terbaik dari [AWS Well-Architected Framework](#) yang membantu pelanggan merancang dan mengoperasikan beban kerja yang andal, aman, efisien, dan hemat biaya di cloud.

Bagian ini menjelaskan bagaimana prinsip-prinsip desain dan praktik terbaik Kerangka Well-Architected diterapkan saat membangun solusi ini.

Keunggulan operasional

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari [pilar keunggulan operasional](#).

- Solusi ini mendorong metrik CloudWatch ke Amazon untuk memberikan pengamatan ke dalam komponennya (seperti infrastruktur dan fungsi Lambda).
- AWS X-Ray melacak fungsi Lambda.
- Menggunakan Amazon SNS untuk pelaporan kesalahan.

Keamanan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik [pilar keamanan](#).

- Semua komunikasi antar layanan menggunakan peran IAM.
- Semua komunikasi multi-akun menggunakan peran IAM.
- Semua peran yang digunakan oleh solusi mengikuti akses hak istimewa paling sedikit. Dengan kata lain, mereka hanya berisi izin minimum yang diperlukan sehingga layanan dapat berfungsi dengan baik.
- Semua penyimpanan data termasuk tabel DynamoDB memiliki enkripsi saat istirahat.

Keandalan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari [pilar keandalan](#).

- Solusinya menggunakan layanan AWS tanpa server sedapat mungkin (seperti Lambda dan DynamoDB) untuk memastikan ketersediaan dan pemulihan yang tinggi dari kegagalan layanan.
- Pemrosesan data menggunakan fungsi Lambda. Solusinya menyimpan data di DynamoDB, sehingga tetap ada di beberapa Availability Zone secara default.

Efisiensi kinerja

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari [pilar efisiensi kinerja](#).

- Solusinya menggunakan arsitektur tanpa server.

- Anda dapat meluncurkan solusi di Wilayah AWS mana pun yang mendukung layanan AWS yang digunakan dalam solusi ini (seperti Lambda dan DynamoDB). Untuk detailnya, lihat [Wilayah AWS yang Didukung](#).
- Solusinya secara otomatis diuji dan digunakan setiap hari. Arsitek solusi dan ahli materi pelajaran kami meninjau solusi untuk area untuk bereksperimen dan meningkatkan.

Optimalisasi biaya

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari [pilar pengoptimalan biaya](#).

- Solusinya menggunakan arsitektur tanpa server, dan pelanggan hanya membayar untuk apa yang mereka gunakan.
- Lapisan komputasi default ke Lambda, yang menggunakan model. pay-per-use

Keberlanjutan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik pilar [keberlanjutan](#).

- Solusinya menggunakan layanan terkelola dan tanpa server untuk meminimalkan dampak lingkungan dari layanan backend.
- Desain tanpa server solusi ini ditujukan untuk mengurangi jejak karbon dibandingkan dengan jejak server lokal yang terus beroperasi.

Tabel konfigurasi penjadwal

Saat diterapkan, Penjadwal Instance di AWS membuat tabel Amazon DynamoDB yang berisi setelan konfigurasi global.

Item konfigurasi global berisi atribut tipe dengan nilai konfigurasi di tabel konfigurasi. Jadwal dan periode berisi atribut tipe dengan nilai jadwal dan periode, masing-masing. [Anda dapat menambahkan, memperbarui, atau menghapus jadwal dan periode dari tabel konfigurasi menggunakan konsol DynamoDB atau antarmuka baris perintah solusi](#). Namun, Anda tidak mengedit item apa pun dengan jenis konfigurasi karena item ini dikelola oleh solusinya.

Penjadwal CLI

Solusinya mencakup CLI yang menyediakan perintah untuk mengkonfigurasi jadwal dan periode. CLI memungkinkan Anda memperkirakan penghematan biaya untuk jadwal tertentu. Perkiraan biaya yang disediakan oleh CLI jadwal hanya untuk tujuan perkiraan. [Untuk informasi selengkapnya tentang mengonfigurasi dan menggunakan CLI penjadwal, lihat CLI Penjadwal.](#)

Layanan AWS yang digunakan dalam solusi ini

AWS service	Deskripsi
AWS Lambda	Inti. Solusi menerapkan fungsi Lambda yang berisi semua logika untuk menjadwalkan instance dan mengelola pembaruan tumpukan menggunakan fitur CloudFormation sumber daya khusus.
Amazon DynamoDB	Inti. Solusi membuat tabel DynamoDB untuk menyimpan konfigurasi jadwal, informasi status, tindakan terakhir yang dilakukan dari instance, dan tabel untuk menyimpan jendela pemeliharaan Systems Manager untuk tujuan penjadwalan.
Amazon CloudWatch	Inti. Solusi menyimpan debugging dan log informasi.
AWS IAM	Inti. Solusi menggunakan IAM untuk mendapatkan izin untuk menjadwalkan instance.
Amazon SNS	Inti. Solusi membuat topik SNS untuk mengirim pesan kesalahan bagi pengguna untuk berlangganan dan memecahkan masalah jika terjadi kesalahan.

AWS service	Deskripsi
AWS KMS	Inti. Solusi membuat kunci AWS KMS untuk mengenkripsi topik SNS.
Amazon EventBridge	Inti. Solusi membuat EventBridge Solusi membuat aturan EventBridge terjadwal yang memanggil AWS lambda pada interval yang konsisten”
AWS Systems Manager	Mendukung. Menyediakan pemantauan sumber daya tingkat aplikasi dan visualisasi operasi sumber daya dan data biaya.
Amazon EC2	Dijadwalkan. Solusi digunakan untuk memulai dan menghentikan instans EC2. Instance diidentifikasi oleh tag tertentu key/values yang dikonfigurasi dalam solusi.
Amazon RDS	Dijadwalkan. Solusi digunakan untuk mengubah status instans RDS DB menjadi Tersedia atau Dihentikan. Instance diidentifikasi oleh tag tertentu key/values yang dikonfigurasi dalam solusi.
Amazon Aurora	Dijadwalkan. Solusi digunakan untuk mengubah status klaster Aurora menjadi Tersedia atau Dihentikan. Cluster diidentifikasi oleh tag tertentu key/values yang dikonfigurasi dalam solusi.
Amazon Neptune	Dijadwalkan. Solusi digunakan untuk mengubah status instans Neptunus menjadi Tersedia atau Berhenti. Instance diidentifikasi oleh tag tertentu key/values yang dikonfigurasi dalam solusi.

AWS service	Deskripsi
Amazon DocumentDB	Dijadwalkan. Solusi digunakan untuk mengubah status instance DocumentDB menjadi Tersedia atau Berhenti. Instance diidentifikasi oleh tag tertentu key/values yang dikonfigurasi dalam solusi.
Grup Penskalaan Otomatis Amazon EC2	Dijadwalkan. Solusi digunakan untuk mengelola aturan penskalaan terjadwal untuk grup EC2 Auto Scaling. Aturan ini akan menjadi grup start/stop Auto Scaling sesuai dengan jadwal terkait. Grup diidentifikasi oleh tag tertentu key/values yang dikonfigurasi dalam solusi.

Keamanan

Saat Anda membangun sistem pada infrastruktur AWS, tanggung jawab keamanan dibagi antara Anda dan AWS. [Model tanggung jawab bersama](#) ini mengurangi beban operasional Anda karena AWS mengoperasikan, mengelola, dan mengontrol komponen termasuk sistem operasi host, lapisan virtualisasi, dan keamanan fisik fasilitas tempat layanan beroperasi. Untuk informasi selengkapnya tentang keamanan AWS, kunjungi [AWS Cloud Security](#).

AWS KMS

Solusi ini membuat kunci terkelola Pelanggan AWS yang dikelola, yang digunakan untuk mengonfigurasi enkripsi sisi server untuk topik SNS dan tabel DynamoDB.

Amazon IAM

Fungsi Lambda solusi memerlukan izin untuk mengakses sumber daya akun hub dan akses ke parameter get/put Systems Manager, akses ke grup CloudWatch log, kunci AWS KMS encryption/decryption, and publish messages to SNS. In addition, Instance Scheduler will also create Scheduling Roles in all managed accounts that will provide access to start/stop EC2, RDS, Resulces Autoscaling, instans DB, memodifikasi atribut instans, dan memperbarui tag untuk sumber daya tersebut. Semua izin yang diperlukan disediakan oleh solusi untuk peran layanan Lambda yang dibuat sebagai bagian dari template solusi.

Saat penerapan Penjadwal Instance akan menerapkan peran IAM cakupan ke bawah untuk setiap fungsi Lambda-nya bersama dengan Peran Penjadwal yang hanya dapat diasumsikan dengan penjadwalan Lambda tertentu di templat hub yang diterapkan. Peran jadwal ini akan memiliki nama mengikuti pola `{namespace}-Scheduler-Role`, dan `{namespace}-ASG-Scheduling-Role`.

Untuk informasi rinci tentang izin yang diberikan untuk setiap peran layanan, lihat [CloudFormation templat](#).

Volume EC2 EBS Terenkripsi

Saat menjadwalkan instans EC2 yang dilampirkan ke volume EBS yang dienkripsi oleh AWS KMS, Anda harus memberikan izin Penjadwal Instans untuk menggunakan kunci AWS KMS terkait. Hal ini memungkinkan Amazon EC2 untuk mendekripsi volume EBS terlampir selama fungsi dimulai. Izin ini harus diberikan kepada peran penjadwalan di akun yang sama dengan instans EC2 menggunakan kunci.

Untuk memberikan izin menggunakan kunci AWS KMS dengan Penjadwal Instance, tambahkan ARN kunci AWS KMS ke tumpukan Penjadwal Instance (hub atau spoke) di akun yang sama dengan instans EC2 menggunakan kunci:

Arn Kunci KMS untuk EC2

Kms Key Arns for EC2

comma-separated list of kms arns to grant Instance Scheduler `kms:CreateGrant` permissions to provide the EC2 service with `Decrypt` permissions for encrypted EBS volumes. This allows the scheduler to start EC2 instances with attached encrypted EBS volumes. provide just (*) to give limited access to all kms keys, leave blank to disable. For details on the exact policy created, refer to security section of the implementation guide (<https://aws.amazon.com/solutions/implementations/instance-scheduler-on-aws/>)

Enter CommaDelimitedList

Ini akan secara otomatis menghasilkan kebijakan berikut dan menambahkannya ke peran penjadwalan untuk akun itu:

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

```
        "kms:ViaService": "ec2.*.amazonaws.com"
    },
    "Null": {
        "kms:EncryptionContextKeys": "false",
        "kms:GrantOperations": "false"
    },
    "ForAllValues:StringEquals": {
        "kms:EncryptionContextKeys": [
            "aws:ebs:id"
        ],
        "kms:GrantOperations": [
            "Decrypt"
        ]
    },
    "Bool": {
        "kms:GrantIsForAWSResource": "true"
    }
},
"Action": "kms:CreateGrant",
"Resource": [
    "Your-KMS-ARNs-Here"
],
"Effect": "Allow"
}
]
}
```

License Manager EC2

Saat menjadwalkan instans EC2 yang dikelola di AWS License Manager, Anda harus memberikan izin Penjadwal Instans untuk menggunakan konfigurasi lisensi terkait. Ini memungkinkan solusi untuk memulai dan menghentikan instance dengan benar sambil mempertahankan kepatuhan lisensi. Izin ini harus diberikan kepada peran penjadwalan di akun yang sama dengan instans EC2 menggunakan License Manager.

Untuk memberikan izin menggunakan AWS License Manager dengan Penjadwal Instance, tambahkan ARN konfigurasi License Manager ke tumpukan Penjadwal Instance (hub atau spoke) di akun yang sama dengan instans EC2 menggunakan License Manager:

License Manager Konfigurasi ARN untuk EC2

License Manager ARNs for EC2

comma-separated list of license manager arns to grant Instance Scheduler ec2:StartInstance permissions to provide the EC2 service with license manager permissions to start the instances. This allows the scheduler to start EC2 instances with license manager configuration enabled. Leave blank to disable. For details on the exact policy created, refer to security section of the implementation guide (<https://aws.amazon.com/solutions/implementations/instance-scheduler-on-aws/>)

Enter CommaDelimitedList

Ini akan secara otomatis menghasilkan kebijakan berikut dan menambahkannya ke peran penjadwalan untuk akun itu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ec2:StartInstances",
      "Resource": [
        "Your-License-Manager-ARNs-Here"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Untuk informasi selengkapnya tentang izin License Manager, lihat [Identitas dan manajemen akses untuk AWS License Manager](#) di Panduan Pengguna AWS License Manager.

Memulai

Panduan ini berisi ikhtisar singkat dan instruksi untuk menyebarkan solusi dengan cepat. Solusi ini menggunakan [CloudFormation templat dan tumpukan AWS](#) untuk mengotomatiskan penerapannya. CloudFormation Template menentukan sumber daya AWS yang disertakan dalam solusi ini dan propertinya. CloudFormation Tumpukan menyediakan sumber daya yang dijelaskan dalam template.

Ikhtisar proses penyebaran

Important

Solusi ini mencakup opsi untuk mengirim metrik operasional anonim ke AWS. Kami menggunakan data ini untuk lebih memahami bagaimana pelanggan menggunakan solusi ini dan layanan serta produk terkait. AWS memiliki data yang dikumpulkan melalui survei ini. Pengumpulan data tunduk pada [Pemberitahuan Privasi](#).

Untuk memilih keluar dari fitur ini, unduh templat, ubah bagian CloudFormation pemetaan AWS, lalu gunakan CloudFormation konsol AWS untuk mengunggah templat Anda yang diperbarui dan menerapkan solusinya.

Ikuti step-by-step petunjuk di bagian ini untuk mengonfigurasi dan menyebarkan solusi ke akun Anda.

Waktu untuk menyebarkan: Sekitar 5-10 menit (tidak termasuk konfigurasi).

[Langkah 1: Luncurkan stack scheduler instance](#)

1. Luncurkan CloudFormation template AWS di akun AWS Anda.
2. Masukkan nilai untuk parameter yang diperlukan.
3. Tinjau parameter template lainnya, dan sesuaikan jika perlu.

[Langkah 2 \(Opsional\): Luncurkan tumpukan jarak jauh di akun sekunder](#)

1. Luncurkan CloudFormation template AWS di akun AWS Anda.
2. Masukkan nilai untuk parameter yang diperlukan.

CloudFormation Templat AWS

Solusi ini menggunakan [CloudFormation templat dan tumpukan AWS](#) untuk mengotomatiskan penerapannya. CloudFormation Template menentukan sumber daya AWS yang disertakan dalam solusi ini dan propertinya. CloudFormation Tumpukan menyediakan sumber daya yang dijelaskan dalam template.

Anda dapat mengunduh CloudFormation templat untuk solusi ini sebelum menerapkannya.

[View template](#)

[scheduler-on-aws](#).template - Gunakan template ini untuk meluncurkan solusi dan semua komponen terkait. Konfigurasi default menerapkan fungsi AWS Lambda, tabel Amazon DynamoDB, acara CloudWatch Amazon, dan metrik khusus CloudWatch , tetapi Anda juga dapat menyesuaikan template berdasarkan kebutuhan spesifik Anda.

[View template](#)

instance-scheduler-on-aws-remote.template - Gunakan template ini untuk meluncurkan peran lintas akun yang digunakan oleh solusi untuk menjadwalkan instance di akun spoke. Untuk penerapan yang menggunakan AWS Organizations, penerapan template juga mendaftarkan akun spoke dengan hub, tidak memerlukan konfigurasi manual.

Note

Jika sebelumnya Anda menerapkan solusi ini, lihat [Memperbarui solusi untuk petunjuk pemutakhiran](#).

Langkah 1: Luncurkan stack hub scheduler instance

Ikuti step-by-step petunjuk di bagian ini untuk menyebarkan solusi ke akun Anda.

Waktu untuk menyebarkan: Sekitar lima menit

[Launch solution](#)

1. Masuk ke [AWS Management Console](#) dan pilih tombol untuk meluncurkan template AWS* instance-scheduler-on-aws .template* AWS. CloudFormation
2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi di Wilayah AWS yang berbeda, gunakan pemilih Wilayah di bilah navigasi konsol.
3. Pada halaman Buat tumpukan, verifikasi bahwa URL templat yang benar ada di kotak teks URL Amazon S3 dan pilih Berikutnya.
4. Pada halaman Tentukan detail tumpukan, tetapkan nama ke tumpukan solusi Anda. Untuk informasi tentang batasan penamaan karakter, lihat [kuota IAM dan AWS STS](#) di Panduan Pengguna AWS Identity and Access Management.
5. Di bawah Parameter, tinjau parameter untuk templat solusi ini dan modifikasi sesuai kebutuhan. Solusi ini menggunakan nilai default berikut.

Parameter	Default	Deskripsi
Infrastruktur		
Namespace	default	Berikan pengenalan unik untuk membedakan antara beberapa penerapan solusi (tanpa spasi). Contoh: Dev.
Gunakan AWS Organizations	No	Gunakan AWS Organizations untuk mengotomatiskan pendaftaran akun spoke.
ID/Remote Akun Organisasi IDs	<Optional Input>	Jika Anda menggunakan AWS Organizations, bidang ini diperlukan. Berikan ID Organisasi, misalnya, o-xxxxyyy . Jika tidak, berikan daftar terpisah koma dari Akun spoke tepercaya IDs yang dapat mendaftarkan diri untuk penjadwalan (maksimum 40), seperti

Parameter	Default	Deskripsi
		1111111111, 2222222222 2
Kunci tag jadwal	Schedule	Kunci tag yang dibaca solusi untuk menentukan jadwal sumber daya. Nilai pada sumber daya menentukan nama jadwal. Jika Anda memilih untuk mengubah nilai default, tetapkan nama yang mudah diterapkan secara konsisten dan benar di semua instance yang diperlukan. Catatan: Kunci tag peka huruf besar/kecil.
Menyimpan data dan log	Enabled	Aktifkan perlindungan penghapusan untuk tabel DynamoDB yang digunakan oleh solusi. Hal ini menyebabkan tabel dipertahankan saat menghapus tumpukan ini. Untuk menghapus tabel saat menghapus tumpukan ini, pertama nonaktifkan parameter ini.
Pengaturan Global		
Aktifkan penjadwalan	Yes	Setel No untuk menanggulangi semua operasi penjadwalan.

Parameter	Default	Deskripsi
Zona waktu default	UTC	Pengidentifikasi zona waktu IANA (International assigned Numbers Authority) default untuk jadwal yang tidak menentukan zona waktu. Untuk daftar pengidentifikasi zona waktu yang valid, lihat kolom pengenalan TZ dari Daftar zona waktu database tz .
Interval penjadwalan (menit)	5	Interval dalam hitungan menit antara penjadwal berjalan. Interval yang lebih pendek meningkatkan akurasi dan daya tanggap tetapi juga meningkatkan biaya. Penyebaran produksi membutuhkan minimal 5 menit untuk operasi yang stabil; nilai yang lebih pendek hanya untuk pengujian skala kecil.
Aktifkan jendela pemeliharaan SSM EC2	No	Izinkan jadwal untuk menentukan satu atau beberapa nama jendela pemeliharaan Systems Manger. Penjadwal Instans di AWS kemudian akan memastikan bahwa instance yang ditandai dengan jadwal tersebut dimulai setidaknya a sepuluh menit sebelum jendela pemeliharaan terkait.

Parameter	Default	Deskripsi
Buat snapshot instance RDS saat berhenti	No	Pilih apakah akan membuat snapshot sebelum menghentikan instans RDS DB. Catatan: Snapshot tidak tersedia untuk klaster Amazon Aurora.
Awalan nama tindakan ASG	IS-	Awalan yang digunakan solusi saat memberi nama tindakan Penskalaan Terjadwal untuk grup Auto Scaling. Tindakan dengan awalan ini akan ditambahkan dan dihapus oleh solusi sesuai kebutuhan.
Kunci tag terjadwal ASG	scheduled	Telah usang. Parameter ini hanya ada untuk tujuan migrasi dan tidak boleh diedit.
Penjadwalan Hub-Account		
Wilayah	<Optional Input>	Daftar Wilayah di mana instance akan dijadwalkan. Misalnya,us-east-1 ,us-west-1 . CATATAN: Jika Anda membiarkan parameter ini kosong, solusinya akan menggunakan Wilayah saat ini.

Parameter	Default	Deskripsi
ARN Kunci KMS untuk EC2	<Optional Input>	Daftar ARN KMS yang dipisahkan koma untuk memberikan Penjadwal Instance di AWS kms: izin untuk menyediakan layanan EC2 CreateGrant izin dekripsi untuk volume EBS terenkripsi. Hal ini memungkinkan scheduler untuk memulai instans EC2 dengan volume EBS terenkripsi terlampir . Menyediakan (*) untuk memberikan akses terbatas ke semua kunci KMS; biarkan kosong untuk menonaktifkan. Untuk detail tentang kebijakan yang dibuat, lihat Volume EBS EC2 Terenkripsi .
License Manager ARN untuk EC2	<Optional Input>	Daftar ARN konfigurasi License Manager yang dipisahkan koma untuk memberikan izin Penjadwal Instance untuk memulai instans EC2 yang dikelola oleh License Manager. Biarkan kosong untuk menonaktifkan. Untuk detailnya, lihat EC2 License Manager .

Pemantauan

Parameter	Default	Deskripsi
Aktifkan CloudWatch Log Debug	No	Aktifkan logging tingkat debug di CloudWatch log.
Periode retensi log (hari)	30	Periode retensi log untuk CloudWatch log dalam beberapa hari.
Pemantauan Operasional	Enabled	Terapkan dasbor wawasan operasional ke CloudWatch dan kumpulkan data metrik khusus tentang pengoperasian solusi. Dasbor dapat dinonaktifkan untuk mengurangi biaya terkait jika diinginkan.

Lainnya

SchedulingRequestHandler Ukuran memori (MB)	512	Ukuran memori fungsi AWS Lambda yang menjadwalkan sumber daya. Tingkatkan jika Anda mengalami penggunaan memori tinggi atau batas waktu.
Ukuran memori orkestrator (MB)	512	Ukuran memori fungsi Lambda orkestrator. Tingkatkan jika Anda mengalami penggunaan memori tinggi atau batas waktu.

6. Pilih Berikutnya.
7. Pada halaman Konfigurasi opsi tumpukan, pilih Berikutnya.
8. Pada halaman Tinjau dan buat, tinjau dan konfirmasi pengaturan. Centang kotak yang mengakui bahwa template akan membuat sumber daya IAM.

9. Pilih Kirim untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status CREATE_COMPLETE dalam waktu sekitar lima menit.

Langkah 2 (Opsional): Luncurkan tumpukan jarak jauh di akun sekunder

Important

Tumpukan jarak jauh harus digunakan di Wilayah yang sama dengan tumpukan hub.

CloudFormation Template AWS otomatis ini mengonfigurasi izin akun sekunder yang memungkinkan tumpukan hub menjadwalkan instance di akun lain. Instal template jarak jauh hanya setelah primary/hub tumpukan berhasil dipasang di akun Hub.

Launch solution

1. Masuk ke AWS Management Console dari akun sekunder yang berlaku dan pilih tombol untuk meluncurkan CloudFormation template AWS instance-scheduler-on-aws -remote.
2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi di Wilayah AWS yang berbeda, gunakan pemilih Wilayah di bilah navigasi konsol. Jika tumpukan hub dikonfigurasi untuk menggunakan AWS Organizations, maka terapkan template jarak jauh di wilayah yang sama dengan tumpukan hub.
3. Pada halaman Buat tumpukan, verifikasi bahwa URL templat yang benar ada di kotak teks URL Amazon S3 dan pilih Berikutnya.
4. Pada halaman Tentukan Detail, tetapkan nama ke tumpukan jarak jauh Anda.
5. Di bawah Parameter, tinjau parameter untuk templat, dan modifikasi.
6. Jika opsi AWS Organizations diaktifkan dan tumpukan hub dikonfigurasi dengan cara yang sama, tidak ada perubahan lebih lanjut yang diperlukan di tumpukan utama untuk memulai penjadwalan.
7. Jika opsi AWS Organization disetel ke Tidak, maka tumpukan hub harus diperbarui dengan ID Akun baru.

Parameter	Default	Deskripsi
Infrastruktur		
Namespace	default	Pengidentifikasi unik yang digunakan untuk membedakan antara beberapa penerapan solusi. Harus disetel ke nilai yang sama dengan tumpukan hub.
Gunakan AWS Organizations	No	Gunakan AWS Organizations untuk mengotomatiskan pendaftaran akun spoke. Harus disetel ke nilai yang sama dengan tumpukan hub.
ID Akun Hub	<Requires Input>	ID Akun Penjadwal Instans di tumpukan hub AWS yang akan menjadwalkan sumber daya di akun ini.
Kunci tag jadwal	Schedule	Kunci tag yang dibaca solusi untuk menentukan jadwal sumber daya. Harus disetel ke nilai yang sama dengan tumpukan hub.
Penjadwalan Akun Anggota		
Wilayah	<Optional Input>	Daftar Wilayah di mana instance akan dijadwalkan. Misalnya, us-east-1, us-west-1. (ini tidak harus daftar wilayah yang sama dengan hub). Jika Anda membiarkan parameter ini kosong, solusinya akan

Parameter	Default	Deskripsi
		menggunakan Wilayah saat ini.
ARN Kunci KMS untuk EC2	<Optional Input>	Daftar ARN KMS yang dipisahkan koma untuk memberikan izin solusi untuk menyediakan layanan EC2 kms: CreateGrant izin dekripsi untuk volume EBS terenkripsi. Hal ini memungkinkan scheduler untuk memulai instans EC2 dengan volume EBS terenkripsi terlampir . Menyediakan (*) untuk memberikan akses terbatas ke semua kunci KMS; biarkan kosong untuk menonaktifkan. Untuk detailnya, lihat Volume EBS EC2 Terenkripsi .
License Manager ARN untuk EC2	<Optional Input>	Daftar ARN konfigurasi License Manager yang dipisahkan koma untuk memberikan izin Penjadwal Instance untuk memulai instans EC2 yang dikelola oleh License Manager. Biarkan kosong untuk menonaktifkan. Untuk detailnya, lihat EC2 License Manager .

1. Pilih Berikutnya.
2. Pada halaman Opsi, pilih Selanjutnya.

3. Pada halaman Tinjau dan buat, tinjau dan konfirmasi pengaturan. Pastikan untuk mencentang kotak yang mengakui bahwa template akan membuat sumber daya IAM.
4. Pilih Kirim untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di AWS CloudFormation Console di kolom Status. Anda akan melihat status CREATE_COMPLETE dalam waktu sekitar lima menit.

Konfigurasi solusinya

Sekarang solusinya telah diterapkan, Anda dapat mulai mengonfigurasi jadwal dan menandai instance untuk penjadwal. Untuk mempelajari lebih lanjut tentang cara melakukan hal-hal ini, lihat [Konfigurasi jadwal](#) dan [contoh Tag untuk penjadwalan](#).

Panduan operator

Panduan ini ditujukan untuk pengguna dan operator solusi ini dan berisi detail tentang cara [mengonfigurasi jadwal](#) dan [memantau solusinya](#).

Konfigurasi jadwal

Setelah solusi berhasil diterapkan, Anda dapat mulai mengonfigurasi jadwal. Penjadwal Instans di AWS mendukung dua metode pengelolaan jadwal seperti yang dijelaskan di bawah ini.

Note

Solusinya dapat mendukung sejumlah jadwal, yang masing-masing dapat berisi satu atau lebih periode yang menentukan kapan instance yang dikendalikan oleh jadwal itu harus berjalan. Untuk informasi lebih lanjut, lihat [Jadwal](#) dan [Periode](#).

Menggunakan Infrastruktur sebagai Kode (disarankan)

Penjadwal Instans di AWS menyediakan AWS CloudFormation CustomResource yang dapat Anda gunakan untuk mengelola jadwal dan periode menggunakan Infrastructure as Code (IaC).

Untuk informasi tentang cara mengelola jadwal menggunakan IaC, silakan lihat [Mengelola Jadwal Menggunakan Infrastruktur sebagai Kode \(IaC\)](#).

Menggunakan Amazon DynamoDB Console dan Instance Scheduler di AWS CLI

Important

Jika Anda menggunakan sumber daya khusus untuk mengelola jadwal apa pun menggunakan IaC, Anda tidak boleh menggunakan konsol DynamoDB atau CLI penjadwal untuk menghapus atau memodifikasi jadwal tersebut atau periodenya. Jika Anda melakukannya, Anda akan membuat konflik antara parameter yang disimpan dalam CloudFormation dan nilai-nilai dalam tabel. Juga, jangan gunakan periode yang dikelola oleh CloudFormation dalam jadwal yang dibuat menggunakan konsol DynamoDB atau CLI penjadwal.

Saat menerapkan Penjadwal Instance di tumpukan hub AWS, solusi tersebut membuat tabel Amazon DynamoDB yang berisi beberapa periode dan jadwal sampel yang dapat Anda gunakan sebagai referensi untuk membuat periode dan jadwal kustom Anda sendiri. Untuk membuat jadwal di DynamoDB, ubah salah satu jadwal di tabel konfigurasi ConfigTable () atau buat yang baru. [Untuk membuat jadwal menggunakan CLI, pertama Instal CLI Scheduler dan kemudian gunakan perintah yang Tersedia.](#)

Note

[Untuk contoh cara membuat beberapa jadwal sampel menggunakan IAc, DynamoDB, dan InstanceScheduler CLI, silakan merujuk ke Jadwal sampel.](#)

Bagian ini memberikan instruksi dan referensi tentang cara menggunakan, memantau dan memperbarui solusi serta pemecahan masalah dan informasi dukungan.

Tandai contoh untuk penjadwalan

Saat menerapkan CloudFormation template AWS, Anda menentukan nama (kunci tag) untuk tag kustom solusi. Untuk Penjadwal Instans di AWS untuk mengenali instans Amazon EC2 atau Amazon RDS, kunci tag pada instance tersebut harus cocok dengan kunci tag khusus ini. Oleh karena itu, penting bagi Anda untuk menerapkan tag secara konsisten dan benar ke semua instance yang berlaku. Anda dapat terus menggunakan [praktik terbaik penandaan](#) yang ada untuk instans Anda saat menggunakan solusi ini. Untuk informasi selengkapnya, lihat [Menandai sumber daya Amazon EC2 Anda dan Menandai sumber daya Amazon RDS](#).

Di AWS Management Console, gunakan [Editor Tag](#) untuk menerapkan atau memodifikasi tag untuk beberapa sumber daya sekaligus. Anda juga dapat menerapkan dan memodifikasi tag secara manual di konsol.

Tak lama setelah menandai sumber daya, ManagedBy tag IS akan diterapkan ke sumber daya oleh Penjadwal Instance untuk menunjukkan bahwa sumber daya sekarang dikelola oleh penjadwal. Anda dapat mencari tag ini untuk mengonfirmasi bahwa sumber daya telah terdaftar dengan benar untuk penjadwalan.

Mengatur nilai tag

Saat Anda menerapkan tag ke instance, gunakan kunci tag yang Anda tentukan selama konfigurasi awal (secara default kunci tag adalah Jadwal) dan atur nilai tag ke nama jadwal yang harus

diterapkan pada instance. Jika Anda ingin mengubah kunci tag, Anda dapat melakukannya dengan [memperbarui parameter solusi](#).

Note

Untuk instans Amazon RDS, nilai tag dapat dari 1 hingga 256 karakter Unicode panjangnya dan tidak dapat diawali dengan aws:. String hanya dapat berisi kumpulan huruf Unicode, digit, spasi putih, '_', ' ', '/', '=', ',', ' - '(Java regex: “^ ([\ p {L}\ p {Z}\ p {N} _.: /=\ -] *) \$”). Untuk informasi selengkapnya, lihat [Menandai sumber daya Amazon RDS](#).

Instans EC2 dengan volume EBS terenkripsi

Jika instans EC2 DB Anda memiliki volume EBS yang dienkripsi dengan kunci KMS yang dikelola pelanggan, Anda harus memberikan peran Penjadwal Instance izin KMS: agar dapat memulai instans tersebut. CreateGrant Untuk informasi selengkapnya, lihat Volume EBS [EC2 Terenkripsi](#).

Instans EC2 dikelola di License Manager

Jika instans EC2 Anda dikelola di AWS License Manager, Anda harus memberikan peran Penjadwal Instance izin License Manager yang sesuai untuk dapat memulai dan menghentikan instans tersebut. Untuk informasi selengkapnya, lihat [EC2 License Manager](#).

Referensi jadwal

Jadwal menentukan kapan instance yang ditandai dengan jadwal itu harus berjalan. Setiap jadwal harus memiliki nama unik, yang digunakan sebagai nilai tag yang mengidentifikasi jadwal yang ingin Anda terapkan ke sumber daya yang ditandai.

Periode

Setiap jadwal harus berisi setidaknya satu periode yang menentukan waktu (s) instance harus dijalankan. Jadwal dapat berisi lebih dari satu periode. Ketika lebih dari satu periode digunakan dalam jadwal, Penjadwal Instans di AWS akan menerapkan tindakan mulai yang sesuai ketika setidaknya salah satu periode benar. Untuk informasi lebih lanjut, lihat [referensi Periode](#).

Zona waktu

Anda juga dapat menentukan zona waktu untuk jadwal. Jika Anda tidak menentukan zona waktu, jadwal akan menggunakan zona waktu default yang Anda tentukan saat Anda meluncurkan solusi. Untuk daftar nilai zona waktu yang dapat diterima, lihat kolom TZ [dari Daftar zona waktu database TZ](#).

Menghentikan bidang instance baru

Bidang `stop_new_instances` mengontrol apakah Penjadwal Instance harus menghentikan instance saat pertama kali diberi tag untuk penjadwalan jika saat ini berjalan di luar periode berjalan. Secara default, bidang ini diatur ke `true`.

Ketika disetel ke `true`, jika Anda menandai instance yang sedang berjalan yang berada di luar periode berjalan yang dijadwalkan, Penjadwal Instance akan segera menghentikan instance tersebut. Ketika disetel ke `false`, Instance Scheduler akan membiarkan instance berjalan hingga waktu berhenti terjadwal berikutnya.

Bidang hibernasi

Bidang hibernasi memungkinkan Anda menggunakan hibernasi untuk instans Amazon EC2 yang dihentikan. Jika bidang ini disetel ke `true`, instans EC2 Anda harus menggunakan Amazon Machine Image (AMI) yang mendukung hibernasi. Untuk informasi selengkapnya, lihat [Linux yang Didukung AMIs](#) di Panduan Pengguna Amazon EC2. Hibernasi menyimpan konten dari memori instans (RAM) ke volume root Amazon Elastic Block Store (Amazon EBS). Jika bidang ini disetel ke `true`, instance dihibernasi alih-alih dihentikan saat solusi menghentikannya.

Jika Anda menyetel solusi untuk menggunakan hibernasi, tetapi instance Anda tidak [dikonfigurasi untuk hibernasi](#) atau tidak memenuhi [prasyarat hibernasi](#), solusi akan mencatat peringatan dan instance dihentikan tanpa hibernasi. Untuk informasi selengkapnya, lihat [Hibernasi Instans Sesuai Permintaan atau Instans Spot Anda](#) di Panduan Pengguna Amazon EC2.

Bidang yang ditegakkan

Jadwal berisi bidang yang diterapkan yang memungkinkan Anda mencegah instance dimulai secara manual di luar periode berjalan, atau dihentikan secara manual selama periode berjalan. Jika bidang ini disetel ke `true` dan pengguna secara manual memulai instance di luar periode berjalan, solusi akan menghentikan instance. Jika bidang ini disetel ke `true`, itu juga memulai ulang instance jika dihentikan secara manual selama periode berjalan.

Pertahankan bidang yang sedang berjalan

Bidang `retain_running` mencegah solusi menghentikan instance di akhir periode berjalan jika instance dimulai secara manual sebelum awal periode. Misalnya, jika instance dengan periode yang berjalan dari jam 9 pagi hingga 5 sore dimulai secara manual sebelum jam 9 pagi, solusi tidak akan menghentikan instance pada jam 5 sore.

Bidang jendela pemeliharaan Systems Manager (hanya berlaku untuk instans EC2)

`ssm-maintenance-window` Bidang ini memungkinkan Anda menambahkan jendela pemeliharaan AWS Systems Manager secara otomatis sebagai periode berjalan ke jadwal. Saat Anda menentukan nama jendela pemeliharaan yang ada di akun dan Wilayah AWS yang sama dengan instans Amazon EC2 Anda, solusi akan memulai instans setidaknya 10 menit sebelum dimulainya jendela pemeliharaan dan menghentikan instance di akhir jendela pemeliharaan jika tidak ada periode berjalan lainnya yang menentukan bahwa instance harus berjalan.

Setelah jendela Pemeliharaan SSM dibuat dan jadwal dikonfigurasi dengan nama jendela pemeliharaan SSM, perubahan diambil pada jadwal berjalan berikutnya dari Lambda. Misalnya, jika Anda memilih frekuensi 5 menit agar Lambda penjadwal berjalan, perubahan jendela pemeliharaan akan diambil oleh Lambda pada interval 5 menit berikutnya.

Penjadwal Instans di AWS akan memastikan bahwa instans Anda dimulai setidaknya 10 menit sebelum jendela pemeliharaan dimulai. Bergantung pada nilai yang Anda tetapkan untuk CloudFormation parameter AWS Interval Penjadwalan, hal ini dapat mengakibatkan instans Anda dimulai 10+menit interval sebelum awal jendela pemeliharaan untuk menjamin bahwa instans dimulai setidaknya 10 menit lebih awal. Misalnya, jika Anda mengatur Interval Penjadwalan menjadi 30 menit, penjadwal akan memulai instance antara 10-40 menit sebelum awal jendela pemeliharaan.

Note

Untuk menggunakan fitur ini, CloudFormation parameter `Enable EC2 SSM Maintenance Windows` di tumpukan hub solusi harus disetel ke `yes`

Untuk informasi selengkapnya, lihat [AWS Systems Manager Maintenance Windows](#) di panduan pengguna AWS Systems Manager.

Tipe instans

Hanya untuk instans Amazon EC2, jadwal memungkinkan Anda menentukan jenis instans opsional yang diinginkan untuk setiap periode dalam jadwal. Saat Anda menentukan jenis instans dalam periode, solusi akan secara otomatis mengubah ukuran instans EC2 agar sesuai dengan jenis instans yang diminta.

<instance-type>Untuk menentukan jenis instance, gunakan sintaks @<period-name>. Misalnya, weekends@t2.nano. Perhatikan bahwa jika Anda menentukan jenis instans untuk periode yang menjadwalkan instans Amazon EC2 dan instans Amazon RDS, jenis instans akan diabaikan untuk instans Amazon RDS.

Jika jenis instance dari instance yang sedang berjalan berbeda dari tipe instance yang ditentukan untuk periode tersebut, solusi akan menghentikan instance yang sedang berjalan dan memulai ulang instance dengan tipe instance yang ditentukan. Untuk informasi selengkapnya, lihat [Mengubah jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Definisi jadwal

Tabel konfigurasi Penjadwal Instance pada AWS di Amazon DynamoDB berisi definisi jadwal. Definisi jadwal dapat berisi bidang-bidang berikut:

Bidang	Deskripsi
description	Deskripsi opsional dari jadwal.
hibernate	Pilih apakah akan hibernasi instans Amazon EC2 yang menjalankan Amazon Linux. Ketika bidang ini disetel ke true, scheduler akan hibernasi instance saat menghentikannya. Perhatikan bahwa instans Anda harus mengaktifkan hibernasi dan harus memenuhi prasyarat hibernasi.
enforced	Pilih apakah akan menegakkan jadwal. Ketika bidang ini disetel ke true, scheduler akan menghentikan instance yang sedang berjalan jika dimulai secara manual di luar periode berjalan atau akan memulai instance jika

Bidang	Deskripsi
	dihentikan secara manual selama periode berjalan.
name	Nama yang digunakan untuk mengidentifikasi jadwal. Nama ini harus unik dan hanya mencakup alfa-numerik, tanda hubung (-), dan garis bawah (_).
periods	<p>Nama periode yang digunakan dalam jadwal ini. Masukkan nama persis seperti yang muncul di bidang nama periode.</p> <p><instance-type>Anda juga dapat menentukan jenis instance untuk periode menggunakan sintaks @<period-name>. Misalnya, weekdays@2.large .</p>
retain_running	Pilih apakah akan mencegah solusi menghentikan instance di akhir periode berjalan jika instance dimulai secara manual sebelum awal periode.
use_maintenance_window	Pilih apakah akan menyertakan jendela pemeliharaan Amazon RDS sebagai periode berjalan ke jadwal instans Amazon RDS, atau jendela pemeliharaan AWS Systems Manager sebagai periode berjalan ke jadwal instans Amazon EC2. Bidang ini diaktifkan secara default dan dapat dinonaktifkan dengan menyetel nilainya ke "false"

Bidang	Deskripsi
<code>ssm_maintenance_window</code>	<p>Pilih apakah akan menambahkan jendela pemeliharaan AWS Systems Manager sebagai periode berjalan tambahan untuk jadwal ini. Menerima nama jendela pemeliharaan yang akan dicocokkan dengan nama jendela sama dengan instans EC2 account/region terjadwal. StringSet</p> <p>Catatan: Fitur ini hanya berlaku untuk instans EC2.</p>
<code>stop_new_instances</code>	<p>Pilih apakah akan menghentikan instance saat pertama kali ditandai jika sedang berjalan di luar periode berjalan. Secara default, bidang ini diatur ke true.</p>
<code>timezone</code>	<p>Zona waktu yang akan digunakan jadwal. Jika tidak ada zona waktu yang ditentukan, zona waktu default (UTC) digunakan. Untuk daftar nilai zona waktu yang dapat diterima, lihat kolom TZ dari Daftar zona waktu database tz.</p>
<code>use_metrics</code>	<p>Pilih apakah akan mengaktifkan CloudWatch metrik di tingkat jadwal. Bidang ini menimpa setelan CloudWatch metrik yang Anda tentukan saat penerapan.</p> <p>Catatan: Mengaktifkan fitur ini akan dikenakan biaya sebesar \$0,90/bulan per jadwal atau layanan terjadwal.</p>

Referensi periode

Periode berisi kondisi yang memungkinkan Anda mengatur jam, hari, dan bulan tertentu yang akan dijalankan instance. Periode dapat berisi beberapa kondisi, tetapi semua kondisi harus benar untuk Penjadwal Instans di AWS untuk menerapkan tindakan mulai atau berhenti yang sesuai.

Mulai dan hentikan waktu

`endtime` dan `begintime` menentukan kapan Penjadwal Instance di AWS akan memulai dan menghentikan instance. Jika Anda menentukan waktu mulai saja, instance harus dihentikan secara manual. Perhatikan bahwa jika Anda menentukan nilai di bidang [hari kerja](#), solusi menggunakan nilai tersebut untuk menentukan kapan harus menghentikan instance. Misalnya, jika Anda menentukan jam 9 pagi tanpa `endtime` dan nilai hari kerja dari Senin hingga Jumat, instans akan dihentikan pada pukul 23:59 pada hari Jumat kecuali Anda telah menjadwalkan periode yang berdekatan.

`begintime`

Demikian pula, jika Anda hanya menentukan waktu berhenti, instance harus dimulai secara manual. Jika Anda tidak menentukan waktu, solusi ini menggunakan aturan hari dalam seminggu, hari dalam sebulan, atau bulan untuk memulai dan menghentikan instance pada beginning/end setiap hari yang sesuai.

Nilai `endtime` dan `begintime` untuk periode Anda harus dalam zona waktu yang ditentukan dalam jadwal. Jika Anda tidak menentukan zona waktu dalam jadwal, solusinya akan menggunakan zona waktu yang ditentukan saat Anda meluncurkan solusi.

Jika jadwal Anda berisi beberapa periode, kami sarankan Anda selalu menentukan `begintime` dan `endtime` periode Anda.

Jika Anda memulai sebuah instance sebelum waktu mulai yang ditentukan, instance akan berjalan hingga akhir periode berjalan. Misalnya, pengguna mungkin menentukan periode yang memulai instance setiap hari pada jam 9 pagi dan menghentikan instance itu pada pukul 5 sore.

9-5 dijadwalkan mulai dan berhenti



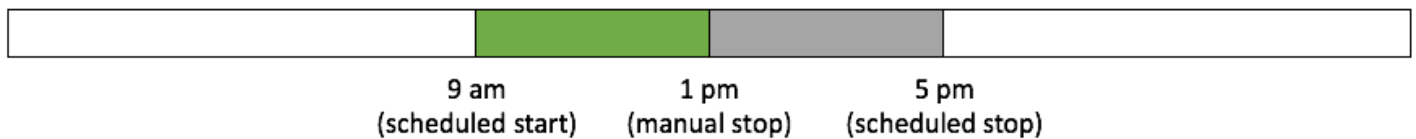
Jika Anda memulai instance itu secara manual pada pukul 5 pagi, solusinya akan menghentikan instance pada jam 5 sore. Jika Anda menggunakan [bidang pertahankan berjalan](#), solusinya tidak akan menghentikan instance pada jam 5 sore.

5 AM dijadwalkan berhenti



Jika Anda menghentikan instance sebelum waktu berhenti yang ditentukan, instance tidak akan berjalan sampai awal periode berjalan berikutnya. Melanjutkan dari contoh sebelumnya, jika pengguna menghentikan instance pada pukul 1 siang pada hari Rabu, solusinya tidak akan memulai instance sampai jam 9 pagi pada hari Kamis.

Garis waktu yang menunjukkan jadwal mulai pukul 9 pagi, pemberhentian manual pada jam 1 siang, dan jadwal berhenti pada jam 5 sore.



Periode yang berdekatan

Solusinya tidak akan berhenti menjalankan instance jika jadwal berisi dua periode berjalan yang berdekatan. Misalnya, jika Anda memiliki jadwal dengan satu periode dengan pukul 11:59 dan periode lain dengan tengah malam pada hari berikutnya, solusinya tidak akan berhenti menjalankan instance, jika tidak ada `weekdaysmonthdays`, atau `months` aturan yang menghentikan instans. `endtime begintime`

Untuk menerapkan jadwal yang menjalankan instance dari jam 9 pagi Senin hingga 5 sore Jumat, solusinya membutuhkan tiga periode. Periode pertama menjalankan instans yang berlaku dari jam 9 pagi hingga 23:59 Senin. Periode kedua berjalan dari tengah malam Selasa hingga 23:59 Kamis. Periode ketiga berjalan dari tengah malam Jumat hingga 5 sore Jumat. Untuk informasi lebih lanjut, lihat [Jadwal sampel](#).

Hari dalam seminggu

`weekdays` Bidang menentukan hari mana selama seminggu sebuah instance akan berjalan. Anda dapat menentukan daftar hari, rentang hari, kejadian n^{th} hari itu dalam sebulan, atau kejadian terakhir hari itu dalam sebulan. Solusinya mendukung nama hari yang disingkat (Mon) dan angka (0).

Hari dalam sebulan

`monthdays`, mendefinisikan hari mana selama bulan sebuah instance akan berjalan. Anda dapat menentukan daftar hari, rentang hari, setiap n^{th} hari dalam sebulan, hari terakhir bulan itu, atau hari kerja terdekat ke tanggal tertentu.

Bulan

`months` Bidang menentukan bulan mana sebuah instance akan berjalan. Anda dapat menentukan daftar bulan, rentang bulan, atau setiap n^{th} bulan. Solusinya mendukung nama bulan yang disingkat (Jan) dan angka (1).

Definisi periode

Tabel konfigurasi Penjadwal Instance pada AWS di Amazon DynamoDB berisi definisi periode. Definisi periode dapat berisi bidang-bidang berikut. Perhatikan bahwa beberapa bidang mendukung karakter [non-standar Cron](#).

Important

Anda harus menentukan setidaknya satu dari item berikut: waktu mulai, waktu akhir, hari kerja, bulan, atau bulan.

Bidang	Deskripsi
<code>begintime</code>	Waktu, dalam format HH: MM, instance akan dimulai.
<code>description</code>	Deskripsi opsional periode tersebut.
<code>endtime</code>	Waktu, dalam format HH: MM, instance akan berhenti.

Bidang	Deskripsi
months	<p>Masukkan daftar bulan yang dibatasi koma, atau rentang bulan dengan tanda hubung, di mana instance akan berjalan. Misalnya, masukkan jan, feb, mar atau 1, 2, 3 jalankan instance selama bulan-bulan tersebut. Atau, Anda bisa masuk jan-mar atau 1-3.</p> <p>Anda juga dapat menjadwalkan instance untuk dijalankan setiap n^{th} bulan atau setiap n^{th} bulan dalam rentang tertentu. Misalnya, masukkan Jan/3 atau 1/3 jalankan instance setiap bulan ketiga mulai bulan Januari. Masuk Jan-Jul/2 untuk berjalan setiap bulan dari Januari hingga Juli.</p>

Bidang	Deskripsi
<p>monthdays</p>	<p>Masukkan daftar hari dalam sebulan yang dibatasi koma, atau rentang hari dengan tanda hubung, di mana instance akan berjalan. Misalnya, masukkan 1, 2, 3 atau 1-3 jalankan instance selama tiga hari pertama setiap bulan. Anda juga dapat memasukkan beberapa rentang. Misalnya, masukkan 1-3, 7-9 untuk menjalankan instance dari 1st ke 3rd dan 7th hingga 9th.</p> <p>Anda juga dapat menjadwalkan instance untuk dijalankan setiap nth hari dalam sebulan atau setiap nth hari dalam satu bulan dalam rentang tertentu. Misalnya, enter 1/7 untuk menjalankan instance setiap hari ketujuh dimulai pada 1st. Masukkan 1-15/2 untuk menjalankan instance setiap hari dari 1st hingga 15th.</p> <p>Masukkan L untuk menjalankan instance pada hari terakhir bulan itu. Masukkan tanggal dan W untuk menjalankan instance pada hari kerja terdekat dengan tanggal yang ditentukan. Misalnya, masukkan 15W untuk menjalankan instance pada hari kerja terdekat ke 15th.</p>
<p>name</p>	<p>Nama yang digunakan untuk mengidentifikasi periode. Nama ini harus unik dan hanya mencakup alfa-numerik, tanda hubung (-), dan garis bawah (_).</p>

Bidang	Deskripsi
weekdays	<p>Masukkan daftar hari dalam seminggu yang dibatasi koma, atau rentang hari dalam seminggu, di mana instance akan berjalan. Misalnya, masukkan <code>0, 1, 2</code> atau <code>0-2</code> jalankan instance Senin hingga Rabu. Anda juga dapat memasukkan beberapa rentang. Misalnya, masukkan <code>0-2, 4-6</code> untuk menjalankan instance setiap hari kecuali Kamis.</p> <p>Anda juga dapat menjadwalkan instance untuk menjalankan setiap kejadian n^{th} dari hari kerja di bulan tersebut. Misalnya, masukkan <code>Mon#1</code> atau <code>0#1</code> jalankan instance pada hari Senin pertama setiap bulan.</p> <p>Masukkan hari dan L untuk menjalankan instance pada kejadian terakhir hari kerja itu di bulan tersebut. Misalnya, masukkan <code>friL</code> atau <code>4L</code> jalankan instance pada hari Jumat terakhir setiap bulan.</p>

Jika suatu periode berisi beberapa kondisi, perhatikan bahwa semua kondisi harus benar untuk Penjadwal Instans di AWS untuk menerapkan tindakan yang sesuai. Misalnya, periode yang berisi weekdays bidang dengan nilai `Mon#1` dan bidang bulan dengan nilai `Jan/3` akan menerapkan tindakan pada hari Senin pertama kuartal tersebut.

Jadwal sampel

Penjadwal Instans di AWS memungkinkan Anda memulai dan menghentikan instans Amazon Elastic Compute Cloud (Amazon EC2) dan Amazon Relational Database Service (Amazon RDS) secara otomatis. Bagian berikut memberikan beberapa contoh jadwal yang dapat disesuaikan dengan banyak kasus penggunaan umum.

Standar 9-5 jam kerja

Jadwal ini menunjukkan cara menjalankan instance pada hari kerja dari jam 9 pagi hingga 5 sore di London.

Periode

Periode ini akan dimulai pada pukul 9 pagi dan menghentikan instans pada jam 5 sore pada hari kerja (Senin-Jumat).

Bidang	Tipe	Nilai
begintime	String	09:00
endtime	String	16:59
name	String	weekdays-9-5
weekdays	StringSet	mon-fri

Jadwal

Nama jadwal memberikan nilai tag yang harus diterapkan pada instance dan zona waktu yang akan digunakan.

Bidang	Tipe	Nilai
name	String	london-working-hours
periods	StringSet	weekdays-9-5
timezone	String	Europe/London

Tag contoh

Untuk menerapkan jadwal ini ke instance, Anda harus menambahkan `Schedule=london-working-hours` tag ke instance. Jika Anda mengubah nama tag default di parameter nama tag AWS CloudFormation Instance Scheduler, tag Anda akan berbeda. Misalnya, jika Anda memasukkan

Sked sebagai nama tag Anda, tag Anda akan menjadi `Sked=london-working-hours`. Untuk informasi selengkapnya, lihat [Menandai sumber daya Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Penjadwal CLI

Untuk mengkonfigurasi jadwal di atas menggunakan [Instance Scheduler CLI](#) gunakan perintah berikut:

```
scheduler-cli create-period --stack <stackname> --name weekdays-9-5 --weekdays mon-fri
--begintime 9:00 --endtime 16:59

scheduler-cli create-schedule --stack <stackname> --name london-working-hours --periods
weekdays-9-5 --timezone Europe/London

Europe/London
```

Sumber daya khusus

CloudFormation Template berikut akan membuat jadwal di atas menggunakan [sumber daya jadwal kustom](#).

Untuk menerapkan template ini, Anda harus menyediakan `ServiceInstanceScheduleServiceTokenARN` yang dapat ditemukan di konsol CloudFormation AWS dengan memilih Stack [Hub Penjadwal Instance yang digunakan sebelumnya](#), lalu pilih Output.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  LondonWorkingWeek:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: london-working-hours
      Description: run instances from 9am to 5pm in London on weekdays
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: Europe/London
```

Periods:

- Description: 9am to 5pm on weekdays
 - BeginTime: '09:00'
 - EndTime: '16:59'
 - WeekDays: mon-fri

Hentikan instance setelah jam 5 sore

Instans dapat dimulai secara bebas kapan saja di siang hari dan jadwal ini akan memastikan bahwa perintah berhenti secara otomatis dikirim kepada mereka pada pukul 5 sore ET setiap hari.

Periode

Periode ini akan menghentikan instance pada jam 5 sore setiap hari.

Bidang	Tipe	Nilai
endtime	String	16:59
name	String	stop-at-5

Jadwal

Nama jadwal memberikan nilai tag yang harus diterapkan pada instance dan zona waktu yang akan digunakan.

Bidang	Tipe	Nilai
name	String	stop-at-5-new-york
periods	StringSet	stop-at-5
timezone	String	America/New York

Tag contoh

Untuk menerapkan jadwal ini ke instance, Anda harus menambahkan `Schedule=stop-at-5-new-york` tag ke instance. Jika Anda mengubah nama tag default di parameter nama tag AWS CloudFormation Instance Scheduler, tag Anda akan berbeda. Misalnya, jika Anda memasukkan Sked

sebagai nama tag Anda, tag Anda akan menjadi `Sked=stop-at-5-new-york`. Untuk informasi selengkapnya, lihat [Menandai sumber daya Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Penjadwal CLI

Untuk mengkonfigurasi jadwal di atas menggunakan [CLI Penjadwal Instance](#), gunakan perintah berikut:

```
scheduler-cli create-period --stack <stackname> --name stop-at-5 --endtime 16:59

scheduler-cli create-schedule --stack <stackname> --name stop-at-5-new-york --periods
stop-at-5 --timezone America/New_York
```

Sumber daya khusus

CloudFormation Template berikut akan membuat jadwal di atas menggunakan [sumber daya jadwal kustom](#).

Untuk menerapkan template ini, Anda harus menyediakan `ServiceInstanceScheduleServiceTokenARN` yang dapat ditemukan di konsol CloudFormation AWS dengan mengklik Stack [Hub Penjadwal Instance yang digunakan sebelumnya](#) dan memilih Output.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  StopAfter5:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: stop-at-5-new-york
      Description: stop instances at 5pm ET every day
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: America/New_York
      Periods:
        - Description: stop at 5pm
          EndTime: '16:59'
```

Hentikan contoh selama akhir pekan

Jadwal ini menunjukkan cara menjalankan instance dari Senin 9 AM ET hingga Jumat 5 PM ET. Karena Senin dan Jumat bukan hari penuh, jadwal ini mencakup tiga periode untuk mengakomodasi: Senin, Selasa-Kamis, dan Jumat.

Periode

Periode pertama mulai menandai instance pada pukul 9 pagi Senin dan berhenti pada tengah malam. Periode ini mencakup bidang dan nilai berikut.

Bidang	Tipe	Nilai
begintime	String	09:00
endtime	String	23:59
name	String	mon-start-9am
weekdays	StringSet	mon

Periode kedua menjalankan instance yang ditandai sepanjang hari Selasa hingga Kamis. Periode ini mencakup bidang dan nilai berikut.

Bidang	Tipe	Nilai
name	String	tue-thu-full-day
weekdays	StringSet	tue-thu

Periode ketiga menghentikan instance yang ditandai pada pukul 5 sore pada hari Jumat. Periode ini mencakup bidang dan nilai berikut.

Bidang	Tipe	Nilai
begintime	String	00:00
endtime	String	16:59

Bidang		Nilai
name	String	fri-stop-5pm
weekdays	StringSet	fri

Jadwal

Jadwal menggabungkan tiga periode ke dalam jadwal untuk instance yang ditandai. Jadwal mencakup bidang dan nilai berikut.

Bidang		Nilai
name	String	senin-9am-jumat-5 sore
periods	StringSet	senin-mulai-9 pagi,, jumat-ber henti-5 sore tue-thu-full-day
timezone	String	Amerika/New_York

Tag contoh

Untuk menerapkan jadwal ini ke instance, Anda harus menambahkan `Schedule=mon-9am-fri-5pm` tag ke instance. Perhatikan bahwa jika Anda mengubah nama tag default di parameter nama tag AWS CloudFormation Instance Scheduler, tag Anda akan berbeda. Misalnya, jika Anda memasukkan `Sked` sebagai nama tag Anda, tag Anda akan menjadi `Sked=mon-9am-fri-5pm`. Untuk informasi selengkapnya, lihat [Menandai sumber daya Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Penjadwal CLI

Untuk mengkonfigurasi jadwal di atas menggunakan [CLI Penjadwal Instance](#), gunakan perintah berikut:

```

scheduler-cli create-period --stack <stackname> --name
mon-start-9am --weekdays mon --begintime 9:00 --endtime 23:59
scheduler-cli create-period --stack <stackname> --name

```

```
tue-thu-full-day --weekdays tue-thu
scheduler-cli create-period --stack <stackname> --namefri-stop-5pm --weekdays fri --
begintime 0:00 --endtime 17:00

scheduler-cli create-schedule --stack <stackname> --name
mon-9am-fri-5pm --periods
mon-start-9am,tue-thu-full-day,fri-stop-5pm -timezone
America/New_York
```

Sumber daya khusus

CloudFormation Template berikut akan membuat jadwal di atas menggunakan [sumber daya jadwal kustom](#).

Untuk menerapkan template ini, Anda harus menyediakan ServiceInstanceScheduleServiceTokenARN yang dapat ditemukan di konsol CloudFormation AWS dengan memilih Stack [Hub Penjadwal Instance yang digunakan sebelumnya](#), lalu pilih Output.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  StopOnWeekends:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: mon-9am-fri-5pm
      Description: start instances at 9am on monday and stop them at 5pm on friday
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: America/New_York
      Periods:
        - Description: 9am monday start
          BeginTime: '09:00'
          EndTime: '23:59'
          WeekDays: mon
        - Description: all day tuesday-thursday
          WeekDays: tue-thu
        - Description: 5pm friday stop
          BeginTime: '00:00'
```

```
EndTime: '16:59'
WeekDays: fri
```

Sumber daya solusi

Sumber daya berikut dibuat sebagai bagian dari Penjadwal Instans di tumpukan AWS.

Nama sumber daya	Tipe	Deskripsi
Utama	AWS::Lambda::Function	Penjadwal Instans pada fungsi AWS Lambda.
Pembantu Konfigurasi Penjadwal	Custom::ServiceSetup	Menyimpan pengaturan konfigurasi global di Amazon DynamoDB.
Izin Pemanggilan Penjadwal	AWS::Lambda::Permission	Mengizinkan CloudWatch peristiwa Amazon menjalankan fungsi AWS Lambda Penjadwal Instans.
Log Penjadwal	AWS::Logs::LogGroup	CloudWatch Grup Log untuk Penjadwal Instance.
Kebijakan Scheduler	AWS::IAM::Policy	Kebijakan yang memungkinkan penjadwal melakukan tindakan mulai dan menghentikan, mengubah atribut instans Amazon EC2, menetapkan tag, dan mengakses sumber daya penjadwal.
Aturan Penjadwal	AWS::Events::Rule	Aturan EventBridge acara Amazon yang memanggil fungsi Lambda penjadwal.
Aturan Peristiwa Metrik Konfigurasi	AWS::Events::Rule	Aturan EventBridge peristiwa Amazon yang secara berkala memanggil fungsi metrik

Nama sumber daya	Tipe	Deskripsi
		anonim deskripsi konfigurasi. Dinonaktifkan saat metrik anonim dinonaktifkan.
Tabel Negara	AWS::Dynamamodb::Table	Tabel DynamoDB yang menyimpan status instance terakhir yang diinginkan.
Tabel Config	AWS::Dynamamodb::Table	Tabel DynamoDB yang menyimpan data konfigurasi, jadwal, dan periode global.
Topik SNS Penjadwal Instance	AWS::SNS::Topic	Mengirim pesan peringatan dan kesalahan ke alamat email berlangganan.

Penjadwal CLI

Penjadwal Instance pada antarmuka baris perintah AWS (CLI) memungkinkan Anda mengonfigurasi jadwal dan periode, serta memperkirakan penghematan biaya untuk jadwal tertentu.

Prasyarat

CLI dalam solusi ini membutuhkan Python 3.8+ dan boto3 versi terbaru.

Kredensial

Untuk menggunakan CLI penjadwal, Anda harus memiliki kredensial untuk AWS CLI. Untuk informasi selengkapnya, lihat [Konfigurasi dan pengaturan file kredensi](#) di Panduan Pengguna AWS CLI.

Kredensi Anda harus memiliki izin berikut:

- `lambda:InvokeFunction`- Untuk menjalankan InstanceSchedulerMain fungsi di tumpukan penjadwal, dan untuk memperbarui informasi jadwal dan periode dalam database konfigurasi penjadwal dari baris perintah
- `cloudformation:DescribeStackResource`- Untuk mengambil ID sumber daya fisik fungsi AWS Lambda dari tumpukan untuk menangani permintaan CLI

Permintaan yang dibuat oleh CLI penjadwal dan tanggapan dicatat di aliran `logAdminCliRequestHandler-yyyymmdd`.

Note

Jika Anda menentukan profil menggunakan argumen nama profil, profil yang Anda tentukan harus memiliki izin ini. Untuk informasi selengkapnya tentang argumen nama profil, lihat Argumen [Umum](#).

Instal CLI Penjadwal

1. [Unduh](#) paket CLI scheduler (`instance_scheduler_cli.zip`) dan letakkan di direktori di komputer Anda.

Important

Instalasi akan gagal jika Anda tidak menempatkan file ke direktori mereka sendiri, dan kemudian menginstalnya dari direktori itu.

2. Buka zip arsip zip ke direktorinya sendiri (`instance_scheduler_cli`).
3. Dari direktori yang sama di mana Anda menempatkan paket CLI yang tidak di-zip, instal `scheduler-cli` ke lingkungan Anda:

Note

Scheduler-cli membutuhkan Python 3.8 atau lebih tinggi dan versi terbaru dari pip dan boto3. Jika Anda tidak memiliki semua ini diinstal pada mesin lokal Anda, silakan merujuk ke [dokumentasi resmi pip](#) untuk petunjuk instalasi sebelum mencoba menginstal Scheduler-CLI.

```
pip install --no-index --find-links=instance_scheduler_cli instance_scheduler_cli
```

4. Verifikasi penginstalan berhasil dengan:

```
scheduler-cli --help
```

Note

Jika disukai, [sdist dari CLI](#) dan dapat diinstal menggunakan proses yang sama seperti di atas.

Struktur perintah

CLI scheduler menggunakan struktur multipart pada baris perintah. Bagian selanjutnya menentukan skrip python CLI scheduler. CLI penjadwal memiliki perintah yang menentukan operasi yang akan dilakukan pada periode dan jadwal. Argumen spesifik untuk operasi dapat ditentukan pada baris perintah dalam urutan apa pun.

```
scheduler-cli <command> <arguments>
```

Argumen umum

CLI scheduler mendukung argumen berikut yang dapat digunakan semua perintah:

Pendapat	Deskripsi
<code>--stack [replaceable]<stackname></code>	Nama tumpukan penjadwal. Penting: Argumen ini diperlukan untuk semua perintah.
<code>--region [replaceable]<regionname></code>	Nama wilayah tempat tumpukan penjadwal digunakan. Catatan: Anda harus menggunakan argumen ini ketika konfigurasi default dan file kredensi tidak diinstal di wilayah yang sama dengan tumpukan solusi.
<code>--profile-name [replaceable]<profilename></code>	Nama profil yang akan digunakan untuk menjalankan perintah. Jika tidak ada nama profil yang ditentukan, profil default digunakan.
<code>--query</code>	JMESPath Ekspresi yang mengontrol output perintah. Untuk informasi selengkapnya tentang

Pendapat	Deskripsi
	mengontrol output, lihat Mengontrol Output Perintah dari Antarmuka Baris Perintah AWS di Panduan Pengguna AWS CLI.
<code>--help</code>	Menampilkan perintah dan argumen yang valid untuk CLI scheduler. Ketika digunakan dengan perintah tertentu, ini menunjukkan sub-perintah dan argumen yang valid untuk perintah itu.
<code>--version</code>	Menunjukkan nomor versi CLI penjadwal.

Perintah yang tersedia

- [menciptakan-periode](#)
- [buat-jadwal](#)
- [hapus-periode](#)
- [hapus-jadwal](#)
- [jelaskan-periode](#)
- [jelaskan-jadwal](#)
- [describe-schedule-usage](#)
- [periode pembaruan](#)
- [jadwal pembaruan-](#)
- [membantu](#)

buat-periode

Deskripsi

Menciptakan periode. Periode harus berisi setidaknya satu dari item berikut: `begintime`, `endtime`, `weekdays`, `months`, atau `monthdays`.

Pendapat

--name

- Nama periode

Tipe: String

Diperlukan: Ya

--description

- Deskripsi periode

Tipe: String

Wajib: Tidak

--begintime

- Waktu ketika periode berjalan dimulai. Jika `begintime` dan tidak `endtime` ditentukan, periode berjalan adalah 00:00 - 23:59.

Tipe: String

Kendala: atau format H:MM HH:MM

Wajib: Tidak

--endtime

- Waktu ketika periode berjalan berhenti. Jika `begintime` dan tidak `endtime` ditentukan, periode berjalan adalah 00:00 - 23:59.

Tipe: String

Kendala: atau format H:MM HH:MM

Wajib: Tidak

--weekdays

- Hari-hari dalam seminggu untuk periode tersebut

Tipe: String

Kendala: Daftar nama hari yang disingkat koma (mon) atau angka (0). Gunakan - untuk menentukan rentang. Gunakan/untuk menentukan setiap nth ^ hari dalam seminggu.

Wajib: Tidak

--months

- Bulan-bulan periode

Tipe: String

Kendala: Daftar nama bulan yang disingkat koma (jan) atau angka (1). Gunakan - untuk menentukan rentang. Gunakan/untuk menentukan setiap nth ^ bulan.

Wajib: Tidak

--monthdays

- Hari-hari dalam sebulan untuk periode tersebut

Tipe: String

Kendala: Daftar nama bulan yang disingkat koma (jan) atau angka (1). Gunakan - untuk menentukan rentang. Gunakan/untuk menentukan setiap nth ^ hari dalam sebulan.

Wajib: Tidak

Contoh

```
$ scheduler-cli create-period --name "weekdays" --begintime 09:00 --endtime 18:00 --
weekdays mon-fri --stack Scheduler
{
  "Period": {
    "Name": "weekdays",
    "Endtime": "18:00",
    "Type": "period",
    "Begintime": "09:00",
    "Weekdays": [
      "mon-fri"
    ]
  }
}
```

buat-jadwal

Deskripsi

Membuat jadwal.

Pendapat

--name

- Nama jadwal

Tipe: String

Diperlukan: Ya

--description

- Deskripsi jadwal

Tipe: String

Wajib: Tidak

--enforced

- Menegakkan status terjadwal untuk contoh

Wajib: Tidak

--use-metrics

- Kumpulkan CloudWatch metrik Amazon

Wajib: Tidak

--periods

- Daftar periode berjalan untuk jadwal. Jika beberapa periode ditentukan, solusi akan memulai sebuah instance jika salah satu periode dievaluasi. true

Tipe: String

Kendala: Daftar periode yang dibatasi koma. Gunakan <period-name>@[replaceable]<instance type> untuk menentukan jenis instance untuk suatu periode. Misalnya, weekdays@2.large.

Wajib: Ya

--retain-running

- Mencegah instance dihentikan oleh solusi pada akhir periode berjalan, jika instance dimulai secara manual sebelum awal periode.

Wajib: Tidak

--ssm-maintenance-window

- Menambahkan jendela pemeliharaan AWS Systems Manager sebagai periode berjalan ke jadwal instans Amazon EC2.

Tipe: String

Wajib: Tidak

--do-not-stop-new-instances

- Jangan menghentikan instance saat pertama kali ditandai jika sedang berjalan di luar periode berjalan

Wajib: Tidak

--timezone

- Zona waktu yang akan digunakan jadwal

Tipe: Array string

Wajib: Tidak (Jika argumen ini tidak digunakan, zona waktu default dari tumpukan solusi utama digunakan.)

--use-maintenance-window

- Menambahkan jendela pemeliharaan Amazon RDS sebagai periode berjalan ke jadwal instans Amazon RDS, atau jendela pemeliharaan AWS Systems Manager sebagai periode berjalan ke jadwal instans Amazon EC2

Jenis: benar/salah

Diperlukan: Tidak (default benar)

Contoh

```
$ scheduler-cli create-schedule --name LondonOfficeHours --periods weekdays,weekends --  
timezone Europe/London --stack Scheduler
```

```
{
  "Schedule": {
    "Enforced": false,
    "Name": "LondonOfficeHours",
    "StopNewInstances": true,
    "Periods": [
      "weekends",
      "weekdays"
    ],
    "Timezone": "Europe/London",
    "Type": "schedule"
  }
}
```

hapus-periode

--name

- Nama periode yang berlaku

Tipe: String

Diperlukan: Ya

Important

Jika periode digunakan dalam jadwal yang ada, Anda harus menghapusnya dari jadwal tersebut sebelum Anda menghapusnya.

Contoh

```
$ scheduler-cli delete-period --name weekdays --stack Scheduler
{
  "Period": "weekdays"
}
```

hapus-jadwal

Deskripsi

Menghapus jadwal yang ada

Pendapat

--name

- Nama jadwal yang berlaku

Tipe: String

Diperlukan: Ya

Contoh

```
$ scheduler-cli delete-schedule --name LondonOfficeHours --stack Scheduler
{
  "Schedule": "LondonOfficeHours"
}
```

jelaskan-periode

Deskripsi

Daftar periode yang dikonfigurasi untuk tumpukan Penjadwal Instance

Pendapat

--name

- Nama periode tertentu yang ingin Anda jelaskan

Tipe: String

Wajib: Tidak

Contoh

```
$ scheduler-cli describe-periods --stack Scheduler
```

```
{
  "Periods": [
    {
      "Name": "first-monday-in-quarter",
      "Months": [
        "jan/3"
      ],
      "Type": "period",
      "Weekdays": [
        "mon#1"
      ],
      "Description": "Every first Monday of each quarter"
    },
    {
      "Description": "Office hours",
      "Weekdays": [
        "mon-fri"
      ],
      "Begintime": "09:00",
      "Endtime": "17:00",
      "Type": "period",
      "Name": "office-hours"
    },
    {
      "Name": "weekdays",
      "Endtime": "18:00",
      "Type": "period",
      "Weekdays": [
        "mon-fri"
      ],
      "Begintime": "09:00"
    },
    {
      "Name": "weekends",
      "Type": "period",
      "Weekdays": [
        "sat-sun"
      ],
      "Description": "Days in weekend"
    }
  ]
}
```

jelaskan-jadwal

Deskripsi

Daftar jadwal yang dikonfigurasi untuk tumpukan Penjadwal Instance.

Pendapat

--name

- Nama jadwal tertentu yang ingin Anda jelaskan

Tipe: String

Wajib: Tidak

Contoh

```
$ scheduler-cli describe-schedules --stack Scheduler
{
  "Schedules": [
    {
      "OverrideStatus": "running",
      "Type": "schedule",
      "Name": "Running",
      "UseMetrics": false
    },
    {
      "Timezone": "UTC",
      "Type": "schedule",
      "Periods": [
        "working-days@t2.micro",
        "weekends@t2.nano"
      ],
      "Name": "scale-up-down"
    },
    {
      "Timezone": "US/Pacific",
      "Type": "schedule",
      "Periods": [
        "office-hours"
      ],
    }
  ]
}
```

```
    "Name": "seattle-office-hours"
  },
  {
    "OverrideStatus": "stopped",
    "Type": "schedule",
    "Name": "stopped",
    "UseMetrics": true
  }
]
```

describe-schedule-usage

Deskripsi

Daftar semua periode yang berjalan dalam jadwal dan menghitung jam penagihan untuk instance. Gunakan perintah ini untuk mensimulasikan jadwal untuk menghitung potensi penghematan, dan periode berjalan setelah membuat atau memperbarui jadwal.

Pendapat

--name

- Nama jadwal yang berlaku

Tipe: String

Diperlukan: Ya

--startdate

- Tanggal mulai periode yang digunakan untuk perhitungan. Tanggal default adalah tanggal saat ini.

Tipe: String

Wajib: Tidak

--enddate

- Tanggal akhir periode yang digunakan untuk perhitungan. Tanggal default adalah tanggal saat ini.

Tipe: String

Wajib: Tidak

Contoh

```
$ scheduler-cli describe-schedule-usage --stack InstanceScheduler --name seattle-office-hours
{
  "Usage": {
    "2017-12-04": {
      "BillingHours": 8,
      "RunningPeriods": {
        "Office-hours": {
          "Begin": "12/04/17 09:00:00",
          "End": "12/04/17 17:00:00",
          "BillingHours": 8,
          "BillingSeconds": 28800
        }
      },
      "BillingSeconds": 28800
    }
  },
  "Schedule": "seattle-office-hours"
```

periode pembaruan

Deskripsi

Memperbarui periode yang ada

Pendapat

update-periodPerintah mendukung argumen yang sama dengan create-period perintah. Untuk informasi lebih lanjut tentang argumen, lihat [perintah create period](#).

Important

Jika Anda tidak menentukan argumen, argumen itu akan dihapus dari periode.

jadwal pembaruan-

Deskripsi

Memperbarui jadwal yang ada

Pendapat

`update-schedule` Perintah mendukung argumen yang sama dengan `create-schedule` perintah. Untuk informasi lebih lanjut tentang argumen, lihat [perintah create schedule](#).

Important

Jika Anda tidak menentukan argumen, argumen itu akan dihapus dari jadwal.

bantuan

Deskripsi

Menampilkan daftar perintah dan argumen yang valid untuk CLI scheduler.

Contoh

```
$ scheduler-cli --help
usage: scheduler-cli [-h] [--version]
                    {create-period,create-schedule,delete-period,delete-
schedule,describe-periods,describe-schedule-usage,describe-schedules,update-
period,update-schedule}
                    ...

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

subcommands:
  Valid subcommands

  {create-period,create-schedule,delete-period,delete-schedule,describe-
periods,describe-schedule-usage,describe-schedules,update-period,update-schedule}
```

```

Commands help
create-period      Creates a period
create-schedule    Creates a schedule
delete-period      Deletes a period
delete-schedule    Deletes a schedule
describe-periods   Describes configured periods
describe-schedule-usage
                   Calculates periods and billing hours in which
                   instances are running
describe-schedules Described configured schedules
update-period      Updates a period
update-schedule    Updates a schedule

```

Ketika digunakan dengan perintah tertentu, `--help` argumen menunjukkan sub-perintah dan argumen yang valid untuk perintah itu.

Contoh perintah khusus

```

$ scheduler-cli describe-schedules --help
usage: scheduler-cli describe-schedules [-h] [--name NAME] [--query QUERY]
                                         [--region REGION] --stack STACK

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Name of the schedule
  --query QUERY         JMESPath query to transform or filter the result
  --region REGION       Region in which the Instance Scheduler stack is
                        deployed
  --stack STACK, -s STACK
                        Name of the Instance Scheduler stack

```

Perbarui pengaturan konfigurasi global

Saat pertama kali menerapkan template Hub Penjadwal Instance di AWS CloudFormation, sejumlah setelan konfigurasi global dipilih sebagai input parameter. Parameter konfigurasi global ini dapat diperbarui kapan saja di dalam CloudFormation konsol.

Untuk memperbarui konfigurasi global Penjadwal Instance, masuk ke yang account/region berisi penerapan hub Anda dan buka konsol CloudFormation AWS. Temukan Stack Hub Penjadwal Instance dan pilih Perbarui → Gunakan Template yang Ada. Perbarui parameter konfigurasi global

yang ingin Anda ubah, lalu pilih next → next → submit untuk melakukan CloudFormation pembaruan sumber daya solusi yang relevan.

Mengelola jadwal menggunakan Infrastructure as Code (IaC)

Important

Terapkan jadwal menggunakan templat terpisah setelah penerapan tumpukan hub selesai.

Penjadwal Instance di AWS menyediakan resource kustom (`ServiceInstanceSchedule`) yang dapat Anda gunakan untuk mengonfigurasi dan mengelola jadwal melalui AWS. CloudFormation Sumber daya kustom menggunakan PascalCase kunci untuk data yang sama dengan tabel konfigurasi Penjadwal Instance di Amazon DynamoDB (lihat contoh templat di bawah ini). Untuk informasi lebih lanjut tentang bidang untuk jadwal, lihat [Definisi Jadwal](#). Untuk informasi lebih lanjut tentang bidang untuk periode, lihat [Definisi Periode](#).

Saat Anda menggunakan sumber daya kustom untuk membuat jadwal, nama jadwal tersebut adalah nama sumber daya logis dari sumber daya kustom secara default. Untuk menentukan nama yang berbeda, gunakan properti `Name` dari sumber daya kustom. Solusinya juga menambahkan nama tumpukan ke nama jadwal sebagai awalan secara default. Jika Anda tidak ingin menambahkan nama tumpukan sebagai awalan, gunakan `NoStackPrefix` properti.

Saat Anda menggunakan `Name` dan `NoStackPrefix` properti, pastikan Anda memilih nama jadwal yang unik. Jika jadwal dengan nama yang sama sudah ada, sumber daya tidak akan dibuat atau diperbarui.

Untuk mulai mengelola jadwal menggunakan IaC, salin dan tempel templat sampel berikut dan sesuaikan sebanyak atau sesedikit jadwal yang Anda inginkan. Simpan file sebagai `file.template` (misalnya: `my-schedules.template`), lalu terapkan template baru Anda menggunakan AWS. CloudFormation Untuk contoh templat jadwal lengkap, lihat [Jadwal Sampel](#).

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
```

```
SampleSchedule1:
  Type: 'Custom::ServiceInstanceSchedule'
  Properties:
    ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN #do not edit this line
    NoStackPrefix: 'False'
    Name: my-renamed-sample-schedule
    Description: a full sample template for creating cfn schedules showing all
possible values
    Timezone: America/New_York
    Enforced: 'True'
    Hibernate: 'True'
    RetainRunning: 'True'
    StopNewInstances: 'True'
    UseMaintenanceWindow: 'True'
    SsmMaintenanceWindow: 'my_window_name'
  Periods:
    - Description: run from 9-5 on the first 3 days of March
      BeginTime: '9:00'
      EndTime: '17:00'
      InstanceType: 't2.micro'
      MonthDays: '1-3'
      Months: '3'
    - Description: run from 2pm-5pm on the weekends
      BeginTime: '14:00'
      EndTime: '17:00'
      InstanceType: 't2.micro'
      WeekDays: 'Sat-Sun'

SampleSchedule2:
  Type: 'Custom::ServiceInstanceSchedule'
  Properties:
    ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN #do not edit this line
    NoStackPrefix: 'True'
    Description: a sample template for creating simple cfn schedules
    Timezone: Europe/Amsterdam
  Periods:
    - Description: stop at 5pm every day
      EndTime: '17:00'
```

Saat menerapkan template, Anda harus menyediakan ServiceToken ARN untuk penerapan Penjadwal Instance di AWS. ARN ini dapat ditemukan di dalamnya CloudFormation dengan menavigasi ke tumpukan Penjadwal Instance yang Anda gunakan, memilih Output, dan mencari ServiceInstanceScheduleServiceToken.

Important

Jangan gunakan konsol DynamoDB atau CLI penjadwal untuk menghapus atau mengubah jadwal dan periode yang dikonfigurasi menggunakan sumber daya khusus. Jika Anda melakukannya, Anda akan membuat konflik antara parameter yang disimpan dalam tumpukan dan nilai-nilai dalam tabel. Selain itu, jangan gunakan periode yang dikonfigurasi menggunakan sumber daya khusus dalam jadwal yang dibuat menggunakan konsol DynamoDB atau CLI penjadwal.

Sebelum menghapus tumpukan Penjadwal Instance utama, Anda harus menghapus semua tumpukan tambahan yang berisi jadwal dan periode yang dibuat menggunakan sumber daya kustom karena tumpukan sumber daya kustom berisi dependensi pada tabel DynamoDB tumpukan utama.

Dalam tabel konfigurasi DynamoDB, jadwal dan periode yang dikonfigurasi dengan sumber daya kustom dapat diidentifikasi dengan atribut `configured_in_stack`. Atribut berisi Nama Sumber Daya Amazon dari tumpukan yang digunakan untuk membuat item.

Menangani Kesalahan Kapasitas Tidak Cukup EC2

Ketika Penjadwal Instance gagal memulai instance karena kapasitas yang tidak mencukupi, perilaku defaultnya adalah mengeluarkan peristiwa yang gagal dimulai (lihat [EventBridge Peristiwa](#)) dan coba lagi pada interval penjadwalan berikutnya. Atau, Penjadwal Instance dapat dikonfigurasi untuk mengubah ukuran instance Anda menjadi tipe instans alternatif sebelum mencoba kembali operasi mulai. Fitur ini membantu meningkatkan ketersediaan instans di lingkungan yang dibatasi kapasitas.

Konfigurasi

Untuk mengaktifkan tipe instans alternatif untuk instans EC2, tambahkan `IS-PreferredInstanceTypes` tag ke instance dengan daftar tipe instans yang dipisahkan koma sesuai urutan preferensi (paling disukai terlebih dahulu):

```
IS-PreferredInstanceTypes: t3.medium,t3.large,m5.large
```

Cara kerjanya

Daftar tipe instance alternatif disediakan sesuai urutan preferensi, dengan tipe pertama menjadi yang paling disukai. Saat Penjadwal Instance mencoba memulai instans EC2:

1. Jika instance saat ini bukan ukuran yang paling disukai, coba ubah ukurannya ke ukuran yang paling disukai sebelum memulai
2. Jika operasi awal berhasil, tidak ada alternatif lebih lanjut yang dicoba
3. Jika operasi awal gagal karena kapasitas yang tidak mencukupi:
 - a. Mencoba mengubah ukuran ke jenis instance alternatif berikutnya dalam daftar
 - b. Mencoba lagi operasi awal
 - c. Jika masih tidak berhasil, coba jenis alternatif berikutnya
 - d. Berlanjut sampai berhasil atau semua alternatif habis

Persyaratan dan pembatasan

Kompatibilitas instans: Jenis instans alternatif harus kompatibel dengan konfigurasi instans saat ini (AMI, subnet, grup keamanan, dll.). Untuk informasi selengkapnya, lihat [Mengubah jenis instans](#) di Panduan Pengguna Amazon EC2.

Format tag: Nilai `IS-PreferredInstanceTypes` tag harus berupa daftar terpisah koma dari jenis instans EC2 yang valid.

Contoh

Untuk instance yang awalnya dikonfigurasi sebagai `t3.small`, Anda dapat mengonfigurasi:

```
Schedule: office-hours
IS-PreferredInstanceTypes: t3.small,t3.medium,t3.large,m5.large
```

Jika `t3.small` instance gagal dimulai karena masalah kapasitas, Penjadwal Instance akan mencoba mengubah ukuran dan memulai instance sebagai `t3.medium`, kemudian `t3.large`, `m5.large` sampai berhasil atau semua opsi habis.

Penjadwalan Grup Penskalaan Otomatis EC2

Penjadwal Instans di AWS mendukung penjadwalan grup Auto Scaling EC2 (ASG) menggunakan Tindakan Penskalaan Terjadwal. Ini berbeda dengan implementasi penjadwalan EC2/RDS dan akan dijelaskan lebih lanjut di bagian ini

Lihat [Penskalaan terjadwal untuk Amazon EC2 Auto Scaling untuk](#) informasi selengkapnya tentang Tindakan Penskalaan Terjadwal.

Ikhtisar penjadwalan ASG

ASGs dapat dijadwalkan dengan menerapkan tag jadwal seperti yang dijelaskan dalam [Penandaan instance](#) untuk penjadwalan

Definisi Running/Stopped untuk ASGs

Ketika Grup Auto Scaling dikonfigurasi, pengguna menentukan kapasitas minimum, yang diinginkan, dan maksimum untuk ASG tersebut. Instance Scheduler mengacu pada nilai-nilai ini sebagai `min-desired-max` atau MDM dari ASG.

Status berjalan dari ASG didefinisikan menggunakan tag `IS-MinDesiredMax` kontrol. Tag ini harus berisi nilai MDM yang diinginkan dalam format `min,desired,max` (misalnya: `1,3,5`).

Jika `IS-MinDesiredMax` tag tidak disediakan ketika ASG pertama kali ditandai untuk penjadwalan, itu akan secara otomatis dihasilkan dari ukuran ASG saat ini pada saat penandaan.

Untuk semua ASGs status berhenti didefinisikan sebagai MDM `0-0-0`.

Perilaku ASG Start/Stop

Ketika Instance Scheduler memulai atau menghentikan ASG, itu mengubah pengaturan kapasitas ASG:

Memulai ASG: Menetapkan kapasitas minimum, yang diinginkan, dan maksimum ke nilai yang ditentukan dalam `IS-MinDesiredMax` tag (atau nilai yang dihasilkan secara otomatis dari saat ASG pertama kali ditandai).

Menghentikan ASG: Mengatur kapasitas minimum, yang diinginkan, dan maksimum menjadi `0-0-0`, yang menyebabkan semua instance di ASG dihentikan.

Batasan

Penjadwalan ASG dilakukan dengan mengonversi Penjadwal Instans pada jadwal AWS menjadi aturan penskalaan terjadwal yang kompatibel dengan layanan ASG. Terjemahan ini bekerja paling baik untuk jadwal periode tunggal sederhana yang tidak menggunakan ekspresi cron yang kompleks.

Fitur jadwal berikut tidak didukung untuk penjadwalan ASG:

- Bendera jadwal lanjutan seperti `diberlakukan` dan `tetap berjalan`.

- Hari kerja ke-N, hari kerja terdekat, dan ekspresi hari kerja terakhir dalam periode.
- Jadwal multi-periode dengan periode yang berdekatan atau tumpang tindih.
 - Saat mengonfigurasi tindakan penskalaan terjadwal untuk jadwal multi-periode, Penjadwal Instans di AWS secara langsung menerjemahkan periode ke start/stop tindakan untuk ASG bahkan ketika periode lain yang tumpang tindih atau berdekatan biasanya akan menyebabkan tindakan tersebut dilewati. beginning/end

Pantau solusinya

Pencatatan dan pemberitahuan

Penjadwal Instance menggunakan pencatatan terstruktur yang dioptimalkan untuk kueri Wawasan CloudWatch Log. Solusi ini mencatat informasi pemrosesan untuk setiap instance yang ditandai, hasil evaluasi periode untuk instance, status instans yang diinginkan selama periode tersebut, tindakan yang diterapkan, dan pesan debugging.

Log ditulis ke Amazon CloudWatch Logs dalam dua grup log:

```
{stackName}-{namespace}-administrative-logs
```

Log untuk pendaftaran dan deregistrasi sumber daya, operasi sumber daya khusus, permintaan CLI, dan kegiatan administratif lainnya.

```
{stackName}-{namespace}-scheduling-logs
```

Log untuk penjadwalan operasi termasuk orkestrasi dan eksekusi penanganan permintaan.

Log peringatan dan kesalahan juga diteruskan ke topik Amazon SNS yang dibuat solusi, yang dapat dikonfigurasi untuk mengirim pesan ke alamat email berlangganan. Anda dapat menemukan nama topik Amazon SNS di tab Output dari tumpukan solusi.

CloudWatch Kueri Wawasan Log

Format logging terstruktur Instance Scheduler memungkinkan kueri yang efisien menggunakan Wawasan CloudWatch Log. Anda dapat menggunakan Wawasan Log untuk mencari, menganalisis, dan memvisualisasikan data log guna memecahkan masalah operasional dan memantau aktivitas penjadwalan.

Penjadwal Instance menyediakan kueri log yang telah diformat sebelumnya yang dapat Anda akses dari bagian Kueri Tersimpan di konsol Log: CloudWatch

SchedulingHistory

Tindakan penjadwalan kueri yang dilakukan pada sumber daya, termasuk memulai dan menghentikan operasi.

RegistrationEvents

Permintaan pendaftaran sumber daya dan acara deregistrasi.

Errors

Log kesalahan kueri untuk memecahkan masalah dengan solusinya.

Untuk informasi selengkapnya tentang Wawasan CloudWatch Log, lihat [Menganalisis data CloudWatch log dengan Wawasan Log](#) di Panduan Pengguna CloudWatch Log Amazon.

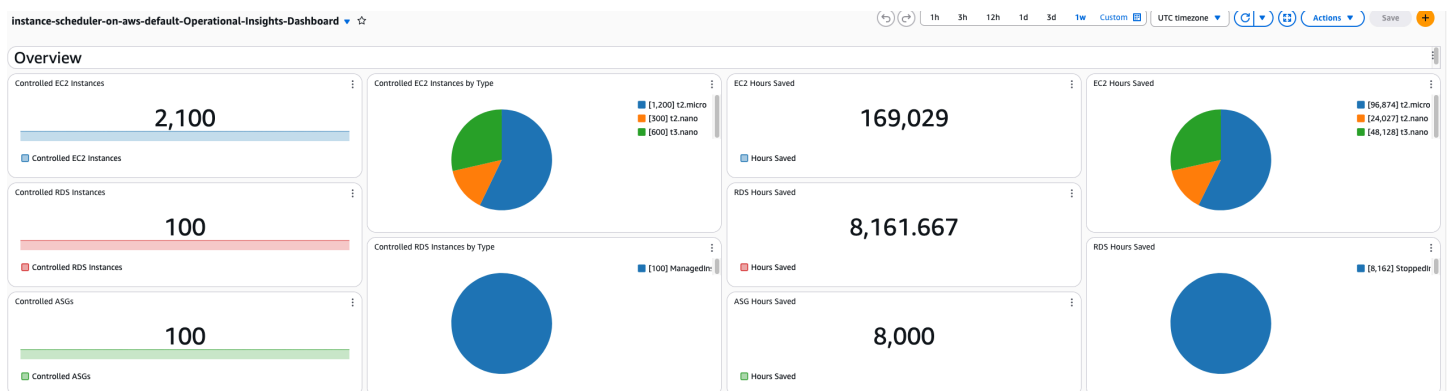
Dasbor wawasan operasional

Dasbor Operational Insights memberikan visibilitas ke dalam kinerja solusi dan penghematan biaya dari manajemen instans terjadwal.

Untuk mengakses dasbor, pastikan Pemantauan Operasional diatur ke “diaktifkan” di parameter tumpukan hub. Arahkan ke CloudWatch dan pilih “Dasbor” dari menu navigasi. Nama dasbornya adalah * {stack-name} -Operational-Insights-Dashboard*.

Dasbor menampilkan jumlah instans terkelola, jam berjalan yang disimpan, dan metrik kinerja fungsi Lambda.

Ikhtisar dasbor wawasan operasional

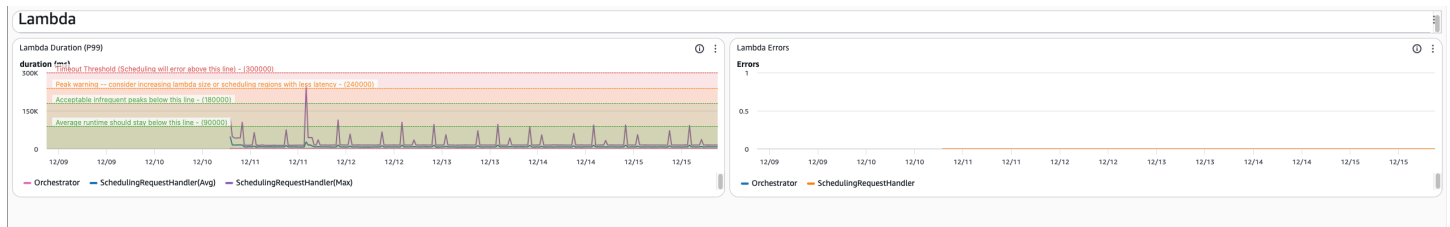


Note

Informasi dalam grafik ini tergantung pada interval penjadwalan yang dikonfigurasi pada tumpukan hub solusi. Saat memperbarui interval penjadwalan solusi, dasbor hanya akan menampilkan metrik penjadwalan dari setelah pembaruan terbaru hingga interval penjadwalan.

Pantau waktu eksekusi Lambda untuk memastikan kinerja optimal (lihat [Kuota](#)). Jika waktu eksekusi secara konsisten mendekati ambang batas waktu, pertimbangkan untuk meningkatkan properti ukuran Lambda atau menerapkan Penjadwal Instance ke wilayah dengan latensi lebih rendah ke wilayah terkelola Anda.

Metrik Lambda menunjukkan durasi dan jumlah kesalahan



Biaya tambahan yang terkait dengan fitur ini

Dasbor operasional ini didukung oleh CloudWatch metrik khusus yang dikumpulkan oleh solusi yang akan dikenakan biaya tambahan. Fitur ini dapat dimatikan dengan menonaktifkan “Pemantauan Operasional” pada tumpukan hub solusi. Fitur ini dikenakan biaya tambahan \$3.00/bulan ditambah biaya penskalaan tambahan berdasarkan ukuran penerapan Anda. Biayanya adalah sebagai berikut:

CloudWatch Dasbor Kustom	\$3
Per-instance-type metrik	\$0,90 per jenis instance*
Penggunaan API	<u>~\$0,10 per target aktif**</u>

*Biaya ini dilacak per kategori layanan (EC2/RDS) dan hanya untuk jenis contoh yang benar-benar digunakan untuk penjadwalan.

*

Monitoring EventBridge Acara

Penjadwal Instans menerbitkan acara penjadwalan dan pendaftaran ke bus EventBridge acara untuk memberikan visibilitas ke dalam operasi solusi dan memungkinkan integrasi dengan layanan AWS lainnya.

Tipe peristiwa

Solusinya menerbitkan dua kategori acara utama:

Acara penjadwalan: Diterbitkan saat Penjadwal Instance mengambil tindakan untuk memulai, menghentikan, atau mengonfigurasi sumber daya terkelola. Peristiwa ini mencakup detail tentang instance, jadwal, dan tindakan yang diambil. mulai, hentikan, atau konfigurasi sumber daya terkelola. Peristiwa ini mencakup rincian tentang contoh, jadwal, dan tindakan yang diambil.

Acara pendaftaran: Diterbitkan ketika sumber daya terdaftar atau tidak terdaftar untuk penjadwalan berdasarkan operasi penandaan.

Tujuan peristiwa

IS- bus LocalEvents acara: Bus `IS-LocalEvents` acara dikerahkan di setiap wilayah yang dikelola dari setiap akun anggota (termasuk akun hub). Setiap bus menerima acara untuk tindakan penjadwalan dan pendaftaran sumber daya di wilayah tersebut.

IS- bus GlobalEvents acara: Bus `IS-GlobalEvents` acara di akun hub menerima salinan dari setiap acara yang dikirim ke bus `IS-LocalEvents` acara apa pun, menyediakan pemantauan terpusat di semua akun dan wilayah.

Menggunakan EventBridge acara

Anda dapat membuat EventBridge aturan untuk:

- Pantau operasi penjadwalan di seluruh infrastruktur Anda
- Memicu pemberitahuan saat instance dimulai atau dihentikan
- Integrasikan dengan layanan AWS lainnya untuk alur kerja otomatis
- Menerapkan pemantauan dan peringatan kepatuhan

Struktur peristiwa

Semua acara menggunakan EventBridge format standar. Contoh berikut menunjukkan struktur untuk setiap jenis acara:

Acara penjadwalan:

```
{
  "Source": "instance-scheduler",
  "DetailType": "Scheduling Action",
  "Resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"],
  "Detail": {
    "account": "123456789012",
    "region": "us-east-1",
    "service": "ec2",
    "resource_id": "i-1234567890abcdef0",
    "requested_action": "Start",
    "action_taken": "Started",
    "schedule": "office-hours"
  }
}
```

Acara pendaftaran:

```
{
  "Source": "instance-scheduler",
  "DetailType": "Resource Registered",
  "Resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"],
  "Detail": {
    "account": "123456789012",
    "region": "us-east-1",
    "service": "ec2",
    "resource_id": "i-1234567890abcdef0",
    "schedule": "office-hours"
  }
}
```

Setiap acara berisi bidang-bidang kunci ini:

- **Source**- Mengidentifikasi sumber acara sebagai “instance-scheduler”
- **DetailType**- Menentukan kategori acara: “Penjadwalan Tindakan” untuk operasi contoh atau “Sumber Daya Terdaftar” untuk menandai peristiwa

- **Resources**- Array yang berisi sumber daya AWS yang terpengaruh ARNs
- **Detail**- Berisi muatan acara dengan ID akun, wilayah, jenis layanan (*ec2/rds*), ID sumber daya, nama jadwal, dan untuk acara penjadwalan, baik tindakan yang diminta maupun hasil aktual

requested_action Nilai yang mungkin untuk acara penjadwalan:

- **Start**: Scheduler dimaksudkan untuk memulai instance
- **Stop**: Scheduler dimaksudkan untuk menghentikan instance
- **Configure**: Scheduler dimaksudkan untuk mengkonfigurasi instance

action_taken Nilai yang mungkin untuk acara penjadwalan:

- **Started**: Instance dimulai
- **Stopped**: Instance dihentikan
- **Hibernated**: Instance hibernasi
- **Configured**: Konfigurasi instans telah dimodifikasi
- **Error**: Terjadi kesalahan selama operasi penjadwalan

Membuat EventBridge aturan

Untuk memantau peristiwa Penjadwal Instance:

1. Arahkan ke EventBridge konsol di akun AWS Anda
2. Buat aturan baru yang menargetkan bus *IS-GlobalEvents* acara (untuk pemantauan terpusat) atau bus *IS-LocalEvents* acara (untuk pemantauan lokal)
3. Tentukan pola acara agar sesuai dengan peristiwa Penjadwal Instance
4. Konfigurasi target seperti topik SNS, fungsi Lambda, atau Log CloudWatch

Untuk informasi lebih lanjut tentang EventBridge, lihat [Apa itu Amazon EventBridge?](#) di Panduan EventBridge Pengguna Amazon.

Pemecahan Masalah

Bagian ini menyediakan instruksi pemecahan masalah untuk menerapkan dan menggunakan solusi.

Resolusi masalah yang diketahui memberikan instruksi untuk mengurangi kesalahan yang diketahui. Jika petunjuk ini tidak mengatasi masalah Anda, [Contact Support](#) memberikan petunjuk untuk membuka kasus Support untuk solusi ini.

Resolusi masalah yang diketahui

Masalah: Instans tidak dijadwalkan di akun jarak jauh (v1.4-v3.0)

Jika Anda melihat instance tidak dijadwalkan di akun jarak jauh.

Resolusi

Perbarui tumpukan hub dengan ID akun sekunder atau selesaikan tugas berikut:

1. Di akun utama, navigasikan ke [CloudWatch konsol](#)
2. Di panel navigasi, pilih Log > Grup Log.
3. Pilih grup log bernama `<STACK_NAME>-logs`
4. Cari aliran log untuk ID Akun (akun jarak jauh).
5. Misalnya, Jika tidak ada aliran log bernama dengan ID akun, buka konsol DynamoDB dan pilih tabel bernama. `<STACK_NAME>-<ConfigTable>-<RANDOM>`
6. Pilih Jelajahi Item dan pilih Jalankan.
7. Pilih jenis item Config.
8. Periksa apakah atribut `remote_account_ids` memiliki ID akun.
9. Periksa apakah ID Akun tidak terlihat di atribut ini.
10. Jika solusi dikonfigurasi ke organisasi aws, hapus instalasi dan instal ulang template jarak jauh di akun jarak jauh.
11. Jika solusi dikonfigurasi untuk menggunakan Akun jarak jauh IDs, perbarui parameter cloudformation Berikan Id Organisasi ATAU Daftar Akun Jarak Jauh IDs dengan daftar akun IDs tempat instance dijadwalkan dan tempat templat jarak jauh digunakan.

Masalah: Contoh tidak dijadwalkan (v3.1 +)

Jika Anda melihat instance tidak dijadwalkan.

Resolusi

1. Verifikasi bahwa sumber daya memiliki ManagedBy tag IS- diterapkan.
2. Jika tag tidak ada, hapus dan buat ulang tag Jadwal untuk memicu ulang pendaftaran.
3. Jika tag masih belum diterapkan, verifikasi wilayah diaktifkan untuk penjadwalan:
 - a. Periksa konfigurasi hub/spoke tumpukan untuk wilayah tersebut, atau
 - b. Arahkan ke [EventBridge konsol](#) di wilayah yang sama dengan sumber daya dan verifikasi bahwa bus peristiwa default memiliki aturan acara dengan awalan IS-tagging.
4. Jika wilayah tidak diaktifkan, perbarui tumpukan Penjadwal Instance untuk menyertakan wilayah dalam CloudFormation parameter region.
5. Jika masalah berlanjut, tinjau [log administrasi solusi](#) untuk kesalahan pendaftaran hub.
6. Konfirmasikan bahwa organisasi Anda tidak memiliki kebijakan yang akan mencegah peristiwa diteruskan dari akun Anda ke akun hub solusi.

Masalah: Instans EC2 terenkripsi tidak dimulai

Instance Scheduler melaporkan bahwa instans EC2 dengan volume EBS terenkripsi sedang dimulai, tetapi tidak pernah benar-benar dimulai.

Resolusi

Lihat Volume [EBS EC2 Terenkripsi untuk mengetahui cara memberikan akses Penjadwal Instance agar dapat menjadwalkan instans EC2 dengan volume EBS terenkripsi](#)

Masalah: Instans RDS tidak berhenti saat Buat Snapshot RDS Diaktifkan

Instans RDS tidak dihentikan dan log penjadwal solusi melaporkan kesalahan (AccessDenied) saat memanggil StopDBInstance operasi karena tidak memiliki izin. `rds:CreateDBSnapshot`

Resolusi

Perbarui solusi ke v3.0.5 atau yang lebih baru atau sebagai alternatif tambahkan `rds:CreateDBSnapshot` izin ke peran penjadwal solusi di setiap akun terjadwal.

Hubungi AWS Support

Jika Anda memiliki [AWS Business Support+](#), [AWS Enterprise Support](#), atau [Unified Operations](#), Anda dapat menggunakan AWS Support Center untuk mendapatkan bantuan ahli terkait solusi ini. Bagian berikut memberikan petunjuk.

Buat kasus

1. Masuk ke [Support Center](#).
2. Pilih Buat kasus.

Bagaimana kami bisa membantu?

1. Pilih Teknis.
2. Untuk Layanan, pilih Solusi.
3. Untuk Kategori, pilih Penjadwal Instance di AWS (Linux atau Windows).
4. Untuk Keparahan, pilih opsi yang paling cocok dengan kasus penggunaan Anda.
5. Saat Anda memasukkan Layanan, Kategori, dan Tingkat Keparahan, antarmuka akan mengisi tautan ke pertanyaan pemecahan masalah umum. Jika Anda tidak dapat menyelesaikan pertanyaan Anda dengan tautan ini, pilih Langkah selanjutnya: Informasi tambahan.

Informasi tambahan

1. Untuk Subjek, masukkan teks yang merangkum pertanyaan atau masalah Anda.
2. Untuk Deskripsi, jelaskan masalah secara rinci, termasuk nama produk ini dan versi yang Anda gunakan, seperti contoh ini: Instance Scheduler on AWS vX.Y.Z.
3. Pilih Lampirkan file.
4. Lampirkan informasi yang dibutuhkan Support untuk memproses permintaan.

Bantu kami menyelesaikan kasus Anda lebih cepat

1. Masukkan informasi yang diminta.
2. Pilih Langkah selanjutnya: Selesaikan sekarang atau hubungi kami.

Selesaikan sekarang atau hubungi kami

1. Tinjau solusi Selesaikan sekarang.
2. Jika Anda tidak dapat menyelesaikan masalah Anda dengan solusi ini, pilih Hubungi kami, masukkan informasi yang diminta, dan pilih Kirim.

Perbarui solusinya

Penjadwal Instance dirancang agar aman untuk diperbarui di tempat menggunakan AWS. CloudFormation Prosedur umum untuk melakukan ini adalah sebagai berikut:

1. Masuk ke [CloudFormation konsol AWS](#), di account/region tempat tumpukan Hub Anda diinstal, pilih `instance-scheduler-on-aws`, dan pilih Perbarui tumpukan.
2. Pilih Buat pembaruan langsung.
3. Pilih Ganti template yang ada.
4. Di bawah Tentukan template:
 - Pilih URL Amazon S3.
 - Salin tautan [templat terbaru](#).
 - Tempel tautan di kotak URL Amazon S3.
 - Verifikasi bahwa URL templat yang benar ditampilkan di kotak teks URL Amazon S3, dan pilih Berikutnya. Pilih Selanjutnya sekali lagi.
5. Di bawah Parameter, tinjau parameter untuk templat dan modifikasi sesuai kebutuhan (lihat daftar perubahan yang melanggar di bawah ini untuk pembaruan parameter yang diperlukan) .Untuk detail tentang masing-masing parameter Untuk detail tentang parameter, lihat [Langkah 1. Luncurkan tumpukan penjadwal instance](#).
6. Pilih Berikutnya.
7. Pada halaman Konfigurasi opsi tumpukan, pilih Berikutnya.
8. Pada halaman Ulasan, tinjau dan konfirmasi pengaturan. Centang kotak yang menyatakan bahwa template akan membuat sumber daya AWS Identity and Access Management (IAM).
9. Pilih Lihat set perubahan dan verifikasi perubahan.
10. Pilih Perbarui tumpukan untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status `UPDATE_COMPLETE` dalam beberapa menit.

Ulangi langkah-langkah di atas untuk `aws-instance-scheduler-remote` tumpukan di setiap akun bicara Anda.

Melanggar Perubahan dalam Versi Tertentu

Saat memperbarui solusi, Anda dapat meningkatkan langsung dari versi lama ke versi yang lebih baru tanpa kehilangan data penting atau gangguan penjadwalan. Silakan lihat di bawah untuk daftar perubahan perilaku dan pelanggaran di setiap versi utama.

Changelog lengkap dapat dilihat [di halaman solusi GitHub](#)

v1.5.0

Versi 1.5.0 menggantikan kebutuhan untuk menyediakan daftar peran penjadwalan lintas akun ARNs dengan kemampuan untuk mengelolanya secara otomatis melalui AWS Organization Anda. Jika Anda tidak ingin menggunakan AWS Organizations, Anda dapat memberikan daftar Akun Spoke IDs dan Penjadwal Instance akan mengelola peran penjadwalan untuk Anda.

Saat memperbarui ke v1.5.0 atau yang lebih baru, Anda harus:

1. Perbarui templat hub menggunakan instruksi pembaruan normal saat memperbarui parameter berikut:
 - a. Pilih namespace unik untuk solusinya.
 - b. Pilih apakah Anda ingin Menggunakan AWS Organizations untuk mengelola pendaftaran spoke ke depan.
 - i. Jika Anda memilih Ya, ganti ID/Remote Akun Organisasi IDs dengan ID Organisasi AWS Anda.
 - ii. Jika Anda memilih Tidak, ganti OrganizationID/ RemoteAccount IDs dengan daftar akun IDs Spoke yang dipisahkan koma.
2. Perbarui semua tumpukan jarak jauh menggunakan instruksi pembaruan normal sambil memperbarui parameter berikut:
 - a. Namespace - sama seperti yang Anda pilih untuk akun hub.
 - b. Gunakan AWS Organizations - sama seperti akun hub.
 - c. ID Akun Hub - ID Akun akun hub (harus tidak berubah dari sebelumnya).

v3.0.0

v3.0.0 Menambahkan dukungan untuk grup penskalaan otomatis EC2 dan memecah fungsi lambda inti solusi menjadi fungsi terpisah dengan tanggung jawab khusus untuk memberikan isolasi

keamanan yang lebih baik untuk setiap fungsi individu. Rilis ini juga memperbarui perilaku log penjadwalan untuk menyertakan log "SchedulingDecision" untuk wawasan yang lebih baik tentang operasi penjadwalan.

V3.0.0 berisi perubahan melanggar berikut dibandingkan dengan versi sebelumnya:

- Fitur "CloudWatch Metrik" di 1.5.x telah diganti dengan Dasbor Wawasan [Operasional](#).
- Metrik per jadwal CloudWatch telah dipindahkan dari. Schedule/Service/MetricName → Schedule/Service/SchedulingInterval/MetricName
- Semua metrik yang ada akan tetap ada, tetapi metrik baru sekarang akan dikumpulkan di bawah namespace baru dan akan tersedia di dasbor solusi.
- ARN kunci KMS untuk digunakan dengan volume EBS terenkripsi pada instans EC2 DB sekarang harus disediakan ke tumpukan di akun masing-masing. hub/spoke CloudFormation (Untuk informasi selengkapnya, lihat Volume [EBS EC2 Terenkripsi](#).)
 - Jika Anda menjadwalkan EC2s dengan Volume EBS Terenkripsi, Anda harus menyalin kunci kunci KMS yang digunakan untuk parameter tumpukan Anda. hub/spoke
- CloudFormation Parameter untuk layanan terjadwal telah dipecah menjadi parameter individual untuk setiap layanan yang didukung.
 - Semua layanan akan diaktifkan secara default dan dapat dinonaktifkan secara individual.
- Instance Scheduler 3.0 tidak kompatibel dengan versi lama dari Instance Scheduler CLI.
 - Anda perlu memperbarui ke versi terbaru dari CLI Penjadwal Instance untuk terus menggunakan perintah CLI.

Selain hal di atas, skema tabel Maintenance Window telah diperbarui dan akan diganti sebagai bagian dari pembaruan. Ini akan mengatur ulang pelacakan untuk jendela pemeliharaan EC2 selama beberapa menit pertama setelah memperbarui ke v3.x dan dalam kasus yang jarang terjadi dapat menyebabkan instance yang saat ini dalam jendela pemeliharaan dihentikan sebelum waktunya segera setelah pembaruan. Setelah data ini telah diregenerasi penjadwalan operasi akan berlanjut seperti biasa.

v3.1.0

v3.1.0 memfaktorkan ulang infrastruktur inti solusi untuk menggunakan peristiwa penandaan AWS untuk melacak kapan sumber daya diberi tag untuk penjadwalan. Pastikan bahwa izin organisasi Anda akan memungkinkan peristiwa penandaan ini dikirim dari akun anggota ke akun hub pusat Anda.

Saat memperbarui ke v3.1.0 atau yang lebih baru:

- Akun Spoke sekarang mendeklarasikan wilayah terjadwal secara independen dari akun hub. Setiap tumpukan spoke harus menentukan wilayah mana yang akan dijadwalkan di akun tersebut menggunakan parameter Wilayah.
- Mode AWS Organizations sekarang diperlukan untuk penerapan dengan lebih dari 40 akun total. Jika Anda memiliki lebih dari 40 akun dan tidak menggunakan mode Organizations, Anda harus mengaktifkannya selama pembaruan.
- Jika instans EC2 dikelola di AWS License Manager yang ingin Anda jadwalkan, tambahkan konfigurasi License Manager ARNs ke ARNs parameter License Manager Configuration di stack Anda hub/spoke CloudFormation. Untuk informasi selengkapnya, lihat [EC2 License Manager](#).
- Solusi akan secara otomatis menerapkan ManagedBy tag IS ke sumber daya setelah diberi tag untuk penjadwalan untuk menunjukkan bahwa mereka dikelola oleh penjadwal.
- Pengubahan ukuran instance terjadwal (mendefinisikan `period-name@size` dalam jadwal) tidak lagi didukung.
- Daftar akun anggota melalui parameter SSM (meneruskan `{param: ssm-param-name}` ke parameter akun di tumpukan hub) tidak lagi didukung. Semua akun tepercaya harus diteruskan ke tumpukan hub pada waktu penerapan.
- Penjadwal Instance akan membutuhkan hingga 6 tag unik pada sumber daya selama penjadwalan. Pastikan kapasitas penandaan yang memadai pada sumber daya bila digabungkan dengan strategi penandaan organisasi Anda yang lain.
- Metrik per jadwal telah dihapus dari CloudWatch.
- Log solusi telah dikemas ulang ke dalam grup log administratif dan penjadwalan yang terpisah dan dioptimalkan untuk kueri dengan Wawasan Log. CloudWatch Silakan merujuk ke [Monitoring the Solution](#) untuk informasi lebih lanjut.
- Start dan stop tag tidak lagi dapat dikonfigurasi melalui CloudFormation parameter. Solusinya sekarang menggunakan nama tag tetap dengan informasi yang lebih kaya untuk melacak tindakan penjadwalan.

Copot pemasangan solusinya

Important

Saat menghapus instalasi solusi, pastikan untuk menghapus semua tumpukan jadwal khusus sebelum menghapus instalasi solusi itu sendiri.

Anda dapat menghapus instans Scheduler pada solusi AWS dari AWS Management Console atau dengan menggunakan AWS Command Line Interface. Untuk menghapus instalasi solusi, hapus tumpukan hub di AWS Cloud Formation bersama dengan semua tumpukan jarak jauh yang diinstal. Anda kemudian dapat menghapus tag penjadwalan apa pun yang telah diterapkan ke instance untuk tujuan penjadwalan.

Note

Jika Protect DynamoDB Tables diaktifkan pada tumpukan hub solusi CloudFormation, akan mempertahankan tabel DynamoDB solusi dan kunci KMS daripada menghapusnya. Jika Anda ingin menghapus sumber daya ini, pastikan properti ini disetel ke Dinonaktifkan sebelum menghapus tumpukan hub. Atau, Anda dapat menghapusnya secara manual setelah tumpukan hub telah dihapus.

Menggunakan Konsol Manajemen AWS

1. Masuk ke [CloudFormation konsol AWS](#).
2. Pada halaman Stacks, pilih tumpukan instalasi solusi ini.
3. Pilih Hapus.

Menggunakan AWS Command Line Interface

Tentukan apakah AWS Command Line Interface (AWS CLI) tersedia di lingkungan Anda. Untuk petunjuk penginstalan, lihat [Apa itu Antarmuka Baris Perintah AWS](#) di Panduan Pengguna AWS CLI. Setelah mengonfirmasi bahwa AWS CLI tersedia, jalankan perintah berikut.

```
$ aws cloudformation delete-stack --stack-name
```

<installation-stack-name>

Panduan developer

Bagian ini menyediakan kode sumber untuk solusi dan daftar bagian yang ditambahkan di sini dan menyertakan tautan ke setiap subtopik.

Kode sumber

Kunjungi [GitHub repositori](#) kami untuk mengunduh file sumber untuk solusi ini dan untuk berbagi penyesuaian Anda dengan orang lain.

Penjadwal Instans pada template AWS dibuat menggunakan [AWS CDK](#). Lihat tautan: File [README.md](#) untuk informasi tambahan.

Referensi

Bagian ini mencakup informasi tentang pengumpulan data, petunjuk ke [sumber daya terkait](#), dan [daftar pembangun](#) yang berkontribusi pada solusi ini.

Pengumpulan data

Solusi ini mengirimkan metrik operasional ke AWS (“Data”) tentang penggunaan solusi ini. Kami menggunakan Data ini untuk lebih memahami bagaimana pelanggan menggunakan solusi ini serta layanan serta produk terkait. Pengumpulan AWS atas Data ini tunduk pada [Pemberitahuan Privasi AWS](#).

Sumber daya terkait

[Resource Scheduler](#) mirip dengan Instance Scheduler di AWS, tetapi implementasinya berbeda dengan cara berikut:

Penjadwal Instans di AWS menggunakan fungsi Lambda untuk sering mengevaluasi jadwal yang disimpan dalam konfigurasinya dan memeriksa apakah instance berada dalam status yang diinginkan. Quicksetup penjadwal sumber daya menggunakan waktu mulai dan berhenti untuk melakukan tindakan mulai dan berhenti menggunakan runbook SSM. Ini terjadi satu kali ketika waktu saat ini sama dengan waktu mulai atau waktu saat ini melewati waktu mulai.

Penjadwal Instans di AWS saat ini memungkinkan penjadwalan untuk Cluster EC2, RDS, dan Aurora. Resource scheduler hanya menjadwalkan atau memulai dan menghentikan instans EC2.

Gunakan Resource scheduler untuk mengidentifikasi instans EC2 dan start/stop mereka pada waktu tertentu.

Gunakan Penjadwal Instance di AWS saat akun harus dipindai secara teratur ke instance. start/stop

Tabel mengidentifikasi solusi mana yang lebih baik berdasarkan skenario.

Skenario	Penjadwal Sumber Daya	Penjadwal Instans di AWS
Jadwalkan instans Amazon Neptunus	Tidak	Ya

Skenario	Penjadwal Sumber Daya	Penjadwal Instans di AWS
Jadwalkan instans Amazon DocumentDB	Tidak	Ya
Jadwalkan instance grup Auto Scaling	Tidak	Ya
Jadwalkan instans EC2	Ya	Ya
Jadwalkan instans RDS	Tidak	Ya
Jadwalkan Cluster Aurora	Tidak	Ya
Mengelola Jadwal dalam Satu Akun (akun hub)	Tidak	Ya
Kelola Jadwal di akun individual	Ya	Tidak
Ubah Integrasi Kalender	Ya	Tidak
Mulai dan Hentikan tindakan saja	Ya	Tidak
Pantau instans secara berkala dan mulai dan berhenti berdasarkan keadaan saat ini	Tidak	Ya

Kontributor

- Arie Leeuwesteijn
- Mahmoud ElZayet
- Ruald Andreae
- Nikhil Reddy
- Caleb Pearson
- Jason DiDomenico

- Max Granat
- Pratyush Das
- Amanda Jones
- Kevin Hargita
- Beomseok Lee
- Abe Wubshet

Revisi

Tanggal publikasi: Oktober 2020

Periksa file [ChangelOG.md](#) di GitHub repositori untuk melihat semua perubahan dan pembaruan penting pada perangkat lunak. Changelog memberikan catatan perbaikan dan perbaikan yang jelas untuk setiap versi.

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili penawaran dan praktik produk AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dari AWS dan afiliasinya, pemasok, atau pemberi lisensinya. Produk atau layanan AWS disediakan “sebagaimana adanya” tanpa jaminan, pernyataan, atau ketentuan dalam bentuk apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

Penjadwal Instans di AWS dilisensikan berdasarkan ketentuan [Lisensi Apache Versi 2.0](#).

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.