



Panduan Implementasi

Penjadwal Instans di AWS



Penjadwal Instans di AWS: Panduan Implementasi

Copyright © 2024 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 properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Ikhtisar solusi	1
Fitur dan manfaat	2
Kasus penggunaan	3
Konsep dan definisi	4
Biaya	4
Contoh harga (bulanan)	5
Didukung Wilayah AWS	10
Penjadwalan instans lintas akun menggunakan akun IDs atau ID Organisasi AWS	11
Mengaktifkan penjadwalan lintas akun menggunakan Akun IDs	11
Mengaktifkan penjadwalan lintas akun menggunakan AWS ID Organisasi	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 EC2 Auto Scaling Amazon	14
Arsitektur	15
Diagram arsitektur	15
AWS Pertimbangan desain Well-Architected	17
Keunggulan operasional	18
Keamanan	18
Keandalan	18
Efisiensi kinerja	19
Optimalisasi biaya	19
Keberlanjutan	19
Tabel konfigurasi penjadwal	20
Penjadwal CLI	20
AWS layanan yang digunakan dalam solusi ini	20
Keamanan	22
AWS KMS	23
Amazon IAM	23
Volume EBS Terenkripsi EC2	23
Memulai	26

Ikhtisar proses penyebaran	26
AWS CloudFormation template	27
Langkah 1: Luncurkan stack hub scheduler instance	27
Langkah 2 (Opsional): Luncurkan tumpukan jarak jauh di akun sekunder	31
Konfigurasi solusinya	33
Panduan operator	34
Konfigurasi jadwal	34
Menggunakan Infrastruktur sebagai Kode (disarankan)	34
Menggunakan Amazon DynamoDB Console dan Instance Scheduler di AWS CLI	34
Tag instance untuk penjadwalan	35
Mengatur nilai tag	35
EC2 instance dengan volume EBS terenkripsi	36
Referensi jadwal	36
Periode	36
Zona waktu	36
Bidang hibernasi	36
Bidang yang ditegakkan	37
Pertahankan bidang yang sedang berjalan	37
Bidang jendela pemeliharaan Systems Manager (hanya berlaku untuk EC2 instance)	37
Jenis instans	38
Definisi jadwal	39
Referensi periode	41
Mulai dan hentikan waktu	41
Hari dalam seminggu	43
Hari dalam sebulan	43
Bulan	43
Definisi periode	43
Penandaan otomatis	46
Jadwal sampel	47
Standar 9-5 jam kerja	48
Hentikan instance setelah jam 5 sore	50
Hentikan contoh selama akhir pekan	52
Sumber daya solusi	55
Penjadwal CLI	56
Prasyarat	56
Kredensial	56

Instal CLI Penjadwal	57
Struktur perintah	58
Argumen umum	58
Perintah yang tersedia	59
menciptakan-periode	60
buat-jadwal	62
hapus-periode	64
hapus-jadwal	65
jelaskan-periode	65
jelaskan-jadwal	67
describe-schedule-usage	68
periode pembaruan	69
jadwal pembaruan-	70
help	70
Perbarui pengaturan konfigurasi global	71
Mengelola jadwal menggunakan Infrastructure as Code (IaC)	72
Fitur lanjutan	74
EC2 Penjadwalan Grup Auto Scaling	74
Pantau solusinya	77
Pencatatan dan pemberitahuan	77
File log	77
Dasbor wawasan operasional	78
Pantau solusinya dengan Service Catalog AppRegistry	80
Kinerja	84
Perbarui solusinya	84
Melanggar Perubahan dalam Versi Tertentu	86
v1.5.0	86
v3.0.0	87
Pemecahan Masalah	88
Resolusi masalah yang diketahui	88
Masalah: Instans tidak dijadwalkan di akun jarak jauh	88
Resolusi	88
Masalah: Pembaruan solusi dari versi apa pun v1.3.x ke v1.5.0	89
Resolusi	89
Masalah: EC2 Instans terenkripsi tidak dimulai	90
Resolusi	90

Masalah: Instans RDS tidak berhenti saat Buat Snapshot RDS Diaktifkan	90
Resolusi	90
Kontak Dukungan	90
Buat kasus	90
Bagaimana kami bisa membantu?	90
Informasi tambahan	91
Bantu kami menyelesaikan kasus Anda lebih cepat	91
Selesaikan sekarang atau hubungi kami	91
Copot pemasangan solusinya	92
Menggunakan AWS Management Console	92
Menggunakan AWS Command Line Interface	92
Panduan pengembang	94
Kode sumber	94
Referensi	95
Pengumpulan data anonim	95
Kuota	97
Kuota untuk AWS layanan dalam solusi ini	97
AWS CloudFormation kuota	98
Sumber daya terkait	98
Kontributor	99
Revisi	101
Pemberitahuan	106
.....	cvii

Otomatiskan instans awal dan penghentian AWS

Tanggal publikasi: Oktober 2020 ([pembaruan terakhir](#): Januari 2025)

AWS Solusi Penjadwal Instans on 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 aktif 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).

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

Panduan implementasi ini memberikan gambaran umum tentang Instance Scheduler tentang AWS solusi, 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 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.	Biaya
Perkiraan biaya untuk menjalankan solusi ini di Wilayah AS Timur (Virginia N.) adalah USD \$13,15 per bulan.	

Jika kau mau.	Baca.
Memahami pertimbangan keamanan untuk solusi ini.	AWS Keamanan yang Dirancang dengan Baik Keamanan
Konfigurasi jadwal.	Tabel konfigurasi penjadwal
Ketahui mana Wilayah AWS yang didukung untuk solusi ini.	Didukung Wilayah AWS
Lihat atau unduh AWS CloudFormation templat yang disertakan dalam solusi ini untuk secara otomatis menyebarkan sumber daya infrastruktur (“tumpukan”) untuk solusi ini.	AWS CloudFormation template
Akses kode sumber dan opsional gunakan AWS Cloud Development Kit (AWS CDK) (AWS CDK) untuk menyebarkan solusi.	GitHubrepositori

Fitur dan manfaat

Jadwal Instans pada AWS solusi 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 aktif 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 AWS CloudFormation Kustom 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 EC2 instans Amazon, Penjadwal Instance aktif 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.

Integrasi dengan Service Catalog AppRegistry dan Application Manager, kemampuan AWS Systems Manager

Solusi ini mencakup AppRegistry sumber daya [Service Catalog](#) untuk mendaftarkan CloudFormation template solusi dan sumber daya dasarnya sebagai aplikasi di Service Catalog AppRegistry dan [Application Manager](#). Dengan integrasi ini, Anda dapat mengelola sumber daya solusi secara terpusat.

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, EC2 instans Amazon 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 AWS istilah, lihat [Daftar Istilah AWS](#).

Biaya

Anda bertanggung jawab atas biaya AWS layanan yang digunakan saat menjalankan Penjadwal Instance pada AWS. Pada revisi terbaru, biaya untuk menjalankan solusi ini penyebaran kecil di dua akun dan dua Wilayah adalah sekitar \$13,15 per bulan. Lihat tabel biaya sampel berikut untuk rincian lebih rinci.

Instance Scheduler on AWS dirancang untuk menjalankan AWS Lambda fungsi beberapa kali per siklus proses. Misalnya, jika Anda menggunakan solusi untuk mengelola instans Amazon EC2 dan Amazon RDS di satu Wilayah untuk dua akun (satu akun tempat solusi diterapkan dan akun lainnya adalah akun silang), solusi tersebut menjalankan lima pemanggilan fungsi Lambda:

- Satu untuk menangani Permintaan Orkestrasi awal dari Event Bridge, yang dipanggil berdasarkan frekuensi yang dipilih (default: lima menit).
- Pemanggilan Lambda tambahan untuk setiap layanan, akun, dan Wilayah.
- Jika [penjadwalan Grup Auto Scaling](#) diaktifkan, pemanggilan orkestrasi lain berjalan untuk semua akun/wilayah setiap jam.

Metrik Operasional Kustom menambahkan biaya tambahan berdasarkan jumlah jadwal dan jenis instans (seperti m2.medium, t3.large) yang dijadwalkan oleh solusi. Jika Anda tidak ingin melacak metrik ini, Anda dapat mematikan fitur ini untuk menghemat biaya. Lihat [dasbor wawasan operasional](#) untuk detail selengkapnya tentang metrik ini dan biaya terkait.

Solusi ini menggunakan penskalaan sesuai permintaan untuk tabel [Amazon DynamoDB](#) untuk memberikan kapasitas baca dan tulis yang memadai.

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

Biaya solusi per proses tergantung pada jumlah instance yang ditandai dan dikelola oleh solusi. Karena jumlah instans DB EC2 dan RDS meningkat, waktu berjalan Lambda juga meningkat secara proporsional.

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

Note

Untuk tujuan pengoptimalan biaya, Penjadwal Instans mengelompokkan semua layanan terkait Amazon RDS ke dalam satu pemanggilan. Jadi, bahkan jika Anda mengaktifkan penjadwalan Amazon RDS, [Amazon Aurora](#), Amazon [Neptunus](#), dan [Amazon DocDB](#); ini masih akan dianggap hanya “RDS” untuk tujuan perhitungan biaya.

Contoh harga (bulanan)

Penyebaran kecil

Contoh penetapan harga ini didasarkan pada asumsi ini:

- Dua akun, dua Wilayah, menjadwalkan semua layanan yang memungkinkan
- 3 jadwal dalam penggunaan aktif
- 20 contoh dari 3 ukuran berbeda
- Interval penjadwalan: 5 menit
- Ukuran fungsi Lambda: 128 MB
- Runtime fungsi Lambda rata-rata: 8 detik

AWS layanan	Dimensi	Biaya Bulanan [USD]
AWS Lambda	288 + 24 penjadwalan berjalan per hari 1+8 fungsi Lambda per run Runtime Lambda rata-rata 8 detik (\$0,0000021/detik) (\$0.0000002/panggilan fungsi Lambda)	~ \$1,50
Metrik Operasional (opsional)	CloudWatch dasbor (\$3/bulan) 3 per-instance-type metrik (\$0,90/bulan) 3 metrik per jadwal* 2 layanan (\$0,60/bulan) ~ 80.000 panggilan/bulan PutMetric (\$0,01/1000)	~ \$10,00
Amazon DynamoDB	~ 75.000 WRU/bulan (\$1,25 per juta) ~ 100.000 RRU/bulan (\$0,5 per juta) biaya penyimpanan yang dapat diabaikan (<\$0,01)	~ \$0,15
AWS KMS	1 AWS KMS kunci (\$1/bulan)	~ \$1,50

AWS layanan	Dimensi	Biaya Bulanan [USD]
	~ 140.000 permintaan API/ bulan (\$0,30/10000)	
Jumlah:		~\$13,15

Penyebaran sedang

Contoh penetapan harga ini didasarkan pada asumsi ini:

- 50 akun, 4 Wilayah, penjadwalan semua layanan yang didukung
- 10 jadwal dalam penggunaan aktif
- 200 contoh dari 10 ukuran berbeda
- Interval penjadwalan: 5 menit
- Ukuran fungsi Lambda: 128 MB
- Runtime Lambda rata-rata: 8 detik
- 5 EC2 Pemeliharaan Windows

AWS layanan	Dimensi	Biaya Bulanan [USD]
AWS Lambda	288 + 24 penjadwalan berjalan per hari 1+400 fungsi Lambda per run Runtime Lambda rata-rata 8 detik (\$0,0000021/detik) (\$0.0000002/panggilan fungsi Lambda)	~ \$64,00
Metrik Operasional (opsional)	CloudWatch dasbor (\$3/bulan) 10 per-instance-type metrik	~ \$60,00

AWS layanan	Dimensi	Biaya Bulanan [USD]
	(\$0,90/bulan) 10 metrik per jadwal* 2 layanan (\$0,60/bulan) ~ 3.5m panggilan/bulan PutMetric (\$0,01/1000)	
Amazon DynamoDB	~ 7m WRU/bulan (\$1.25 per juta) ~ 8m RRU/bulan (\$0.5 per juta) Biaya penyimpanan (<\$0,01)	~ \$12.00
AWS KMS	1 AWS KMS kunci (\$1/bulan) ~ 7 juta permintaan API/bulan (\$0.30/10000)	~ \$22,00
Jumlah:		~ \$158,00

Penyebaran besar

Contoh penetapan harga ini didasarkan pada asumsi ini:

- 120 akun, 6 Wilayah, penjadwalan Amazon EC2 dan Amazon RDS
- 100 jadwal dalam penggunaan aktif
- 2000 contoh dari 50 ukuran berbeda
- 100 EC2 Pemeliharaan Windows
- Interval penjadwalan: 5 menit
- Ukuran fungsi Lambda: 128 MB
- Runtime fungsi Lambda rata-rata: 8 detik

AWS layanan	Dimensi	Biaya Bulanan [USD]
AWS Lambda	<p>288 + 24 penjadwalan berjalan per hari</p> <p>1+1440 fungsi Lambda per run</p> <p>Runtime fungsi Lambda rata-rata 8 detik</p> <p>(\$0,0000021/detik)</p> <p>(\$0.0000002/Panggilan Lambda)</p>	~\$230,00
Metrik Operasional (opsional)	<p>CloudWatch dasbor (\$3/bulan)</p> <p>50 per-instance-type metrik (\$0,90/bulan)</p> <p>100 metrik per jadwal* 2 layanan (\$0,60/bulan)</p> <p>~ 3.5m panggilan/bulan PutMetric (\$0,01/1000)</p>	~ \$300,00
Amazon DynamoDB	<p>~ 26m WRU/bulan (\$1,25 per juta)</p> <p>~ 26 juta RRU/bulan (\$0.5 per juta)</p> <p>Biaya penyimpanan (<\$0,01)</p>	~ \$40,00
AWS KMS	<p>Kunci 1 KMS (\$1/bulan)</p> <p>~ 25 juta permintaan API/bulan (\$0.30/10000)</p>	~ \$80,00

AWS layanan	Dimensi	Biaya Bulanan [USD]
Jumlah:		~ \$650,00

Untuk mengonfigurasi solusi secara efisien, pertimbangkan hal berikut:

1. Terapkan solusi di Wilayah di mana biaya fungsi Lambda terendah.
2. Jangan mengubah memori fungsi Lambda (CloudFormation parameter Memori kecuali benar-benar diperlukan). Ini akan meningkatkan biaya solusi secara signifikan.
3. Hapus jadwal yang tidak digunakan dari konfigurasi solusi.
4. Pilih frekuensi yang mengurangi jumlah fungsi Lambda berjalan per hari. Misalnya, jika jadwal terpisah jam, atur frekuensi (CloudFormation parameter Frekuensi) ke kenaikan satu jam. Secara default, solusinya diatur ke lima menit yang berarti fungsi Lambda akan dipanggil 288 kali sehari, sedangkan frekuensi satu jam akan berjalan 24 kali sehari.

Didukung Wilayah AWS

Anda dapat menerapkan Penjadwal Instance di wilayah mana pun Wilayah AWS, 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 DB yang diberi tag EC2 dan RDS 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 Instance pada AWS tindakan memengaruhi instance 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 instance, 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

Untuk Wilayah keikutsertaan, Penjadwal Instance on AWS dapat menargetkan instans dalam Wilayah keikutsertaan apa pun untuk penjadwalan, tetapi CloudFormation tumpukan itu sendiri saat ini hanya tersedia untuk penerapan di Wilayah keikutsertaan berikut.

Penjadwalan instans lintas akun menggunakan akun IDs atau ID Organisasi AWS

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 [instance-scheduler-on-aws](#) AWS CloudFormation template 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 ID Organisasi

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

1. Masuk ke [AWS Management Console](#) dan pilih tombol untuk meluncurkan [instance-scheduler-on-aws](#) AWS CloudFormation template di akun utama.
2. Atur 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 Instance pada tumpukan AWS hub di akun yang sama dengan penyimpanan parameter Anda.

Layanan yang didukung untuk penjadwalan

Penjadwal Instance AWS saat ini mendukung penjadwalan layanan berikut:

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

Perilaku shutdown instance

Amazon EC2

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

Secara default, EC2 instance dikonfigurasi untuk berhenti, bukan mengakhiri, saat dimatikan, tetapi Anda dapat [mengubah perilaku ini](#). Oleh karena itu, pastikan bahwa instance yang Anda kontrol menggunakan Penjadwal Instance AWS dikonfigurasi dengan perilaku Stop shutdown; jika tidak, mereka 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 AWS CloudFormation template Create RDS Instance Snapshot untuk membuat snapshot instans RDS DB sebelum solusi menghentikan instance. Snapshot disimpan hingga waktu berikutnya instance dihentikan dan snapshot baru dibuat.

Note

Snapshot tidak tersedia untuk cluster 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 EC2 Auto Scaling Amazon

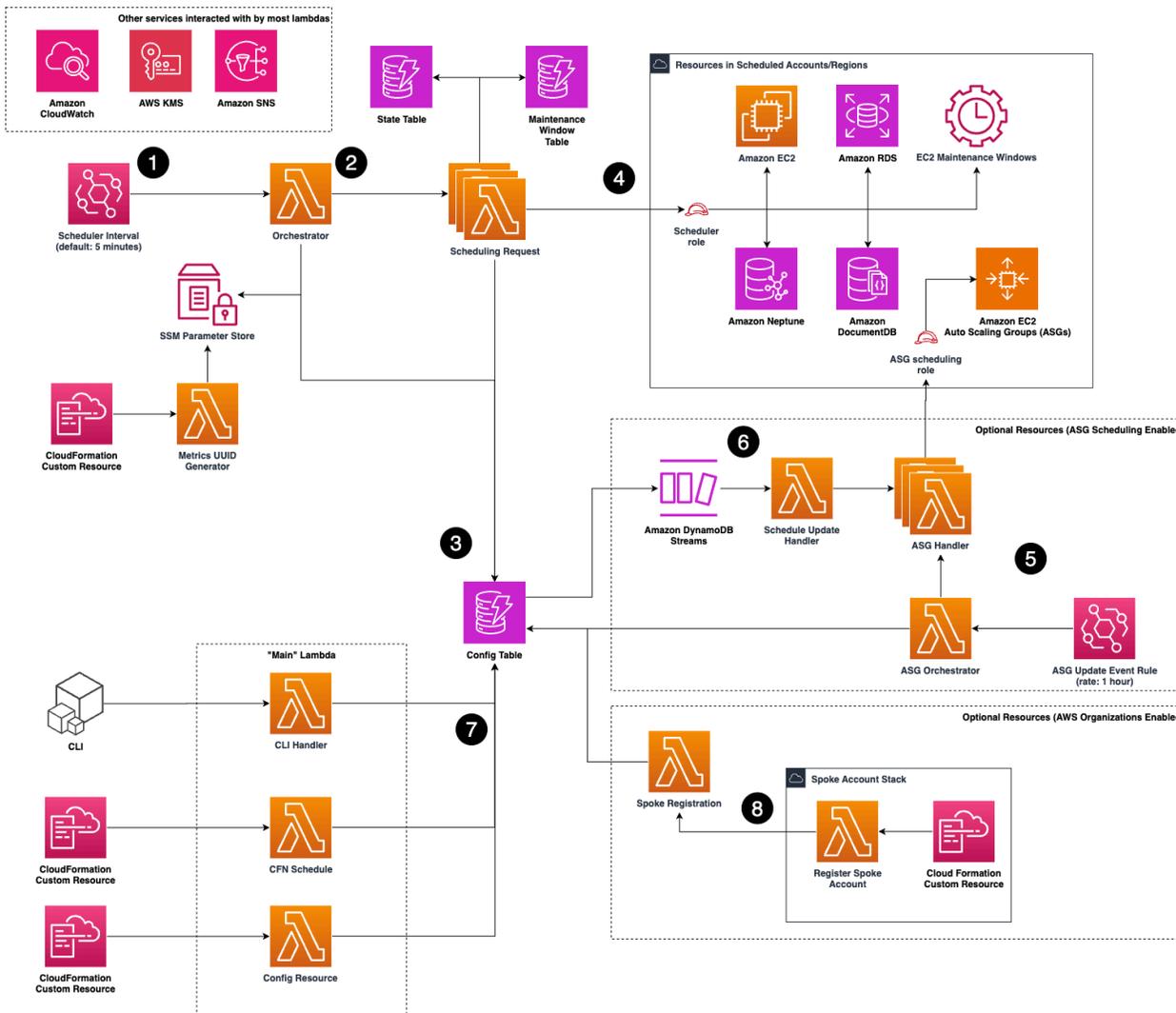
Kami merancang solusi ini untuk menghentikan grup EC2 Auto Scaling Amazon secara otomatis dengan menggunakan tindakan penskalaan terjadwal. Anda dapat menggunakan solusi untuk mengonfigurasi tindakan penskalaan terjadwal pada grup Auto Scaling (ASG). Ketika ASG dihentikan oleh tindakan penskalaan terjadwal, 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 yang digunakan dalam solusi ini. [AWS](#)

Diagram arsitektur

Menerapkan solusi ini akan menyebarkan komponen berikut di file Anda. Akun AWS



Penjadwal Instance di Cloud AWS

1. Solusinya menerapkan [EventBridge Aturan Amazon](#) dengan interval penjadwalan yang dapat dikonfigurasi. Interval penjadwalan ini menentukan seberapa sering solusi berjalan dan mengambil tindakan untuk menjadwalkan instans Anda.
2. Setiap interval penjadwalan memanggil fungsi [AWS Lambda](#) Orkestrasi. Ini akan menentukan daftar Akun AWS, Wilayah, dan layanan yang perlu dijadwalkan. Interval penjadwalan kemudian akan memanggil beberapa fungsi Lambda permintaan penjadwalan secara paralel untuk melakukan aktivitas penjadwalan.
3. Kumpulan jadwal dan periode disimpan dalam tabel konfigurasi [Amazon DynamoDB](#) untuk mengontrol perilaku penjadwalan solusi. Anda dapat mengonfigurasi sejumlah jadwal/periode dalam tabel ini dan solusinya akan menjadwalkan instance yang sesuai.
4. Setiap permintaan penjadwalan memeriksa sumber daya dalam target tertentu (account/Region/service) untuk menemukan sumber daya yang telah ditandai untuk penjadwalan dengan menggunakan jadwal yang ditentukan dalam tabel konfigurasi solusi. Penangan permintaan penjadwalan kemudian memeriksa jadwal yang dikonfigurasi dan melakukan tindakan penjadwalan yang diperlukan.
5. Jika penjadwalan ASG diaktifkan, Penjadwal Instans saat AWS menerapkan [EventBridge aturan Amazon](#) per jam serta sumber daya orkestrasi serta penanganan terkait untuk mengelola Tindakan Penskalaan Terjadwal untuk grup Penskalaan Otomatis [Amazon yang telah diberi tag untuk penjadwalan oleh EC2 solusi](#).
6. Selain pemindaian per jam, solusinya juga melacak pembaruan jadwal di tabel konfigurasi. Ketika jadwal diperbarui, fungsi Lambda orkestrasi sekunder dipanggil untuk memastikan bahwa tindakan penskalaan terjadwal ASG tetap up to date dengan konfigurasi jadwal terbaru.
7. Solusi ini menyediakan beberapa cara untuk membuat/memperbarui jadwal dalam tabel konfigurasi solusi, bersama dengan beberapa contoh jadwal untuk digunakan sebagai titik awal. [Metode konfigurasi meliputi: Konsol DynamoDB, CLI penjadwal, dan sumber daya Kustom.AWS CloudFormation](#)
8. Jika Mode AWS Org diaktifkan dan ID Org yang valid diberikan saat solusi diterapkan, Penjadwal Instance on AWS secara otomatis mendaftarkan tumpukan spoke yang baru diterapkan dengan tumpukan hub solusi. [Tumpukan hub dan spoke harus digunakan di Wilayah yang sama dan di akun yang merupakan anggota Organisasi yang sama AWS](#).

Note

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

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 (ditentukan 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 DB yang dihentikan EC2 atau RDS 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 EC2 instans Amazon berbeda dengan menghentikan EC2 instans Amazon. Secara default, EC2 instans Amazon 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.

AWS Pertimbangan desain 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 dari [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](#).

- Solusi ini menggunakan AWS layanan 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 dalam apa pun Wilayah AWS yang mendukung AWS layanan yang digunakan dalam solusi ini (seperti Lambda dan DynamoDB). Untuk detailnya, lihat [Didukung Wilayah AWS](#).
- 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 di-deploy, Instance Scheduler on 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 solusi.

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.](#)

AWS layanan yang digunakan dalam solusi ini

AWS layanan	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 layanan	Deskripsi
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 KMS	Inti. Solusi membuat AWS KMS kunci 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 EC2 instance. Instance diidentifikasi oleh kunci/nilai tag tertentu yang dikonfigurasi dalam solusi.
Amazon RDS	Dijadwalkan. Solusi digunakan untuk mengubah status instans RDS DB menjadi Tersedia atau Dihentikan. Instance diidentifikasi oleh kunci/nilai tag tertentu yang dikonfigurasi dalam solusi.

AWS layanan	Deskripsi
Amazon Aurora	<p>Dijadwalkan. Solusi digunakan untuk mengubah status klaster Aurora menjadi Tersedia atau Dihentikan. Cluster diidentifikasi oleh tag kunci/nilai tertentu yang dikonfigurasi dalam solusi.</p>
Amazon Neptune	<p>Dijadwalkan. Solusi digunakan untuk mengubah status instans Neptunus menjadi Tersedia atau Berhenti. Instance diidentifikasi oleh kunci/nilai tag tertentu yang dikonfigurasi dalam solusi.</p>
Amazon DocumentDB	<p>Dijadwalkan. Solusi digunakan untuk mengubah status instance DocumentDB menjadi Tersedia atau Berhenti. Instance diidentifikasi oleh kunci/nilai tag tertentu yang dikonfigurasi dalam solusi.</p>
Grup EC2 Auto Scaling Amazon	<p>Dijadwalkan. Solusi digunakan untuk mengelola aturan penskalaan terjadwal untuk grup EC2 Auto Scaling. Aturan-aturan ini akan start/stop Auto Scaling groups in accordance with an associated schedule. Groups are identified by specific tags key/values yang dikonfigurasi dalam solusi.</p>

Keamanan

Ketika Anda membangun sistem di atas AWS infrastruktur, 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 lebih lanjut tentang AWS keamanan, kunjungi [AWS Cloud Keamanan](#).

AWS KMS

Solusinya membuat kunci AWS terkelola Pelanggan terkelola, 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 get/put Systems Manager parameters, access to CloudWatch log groups, AWS KMS key encryption/decryption, and publish messages to SNS. In addition, Instance Scheduler on AWS 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 di AWS akan menerapkan peran IAM tercakup ke bawah untuk setiap fungsi Lambda beserta Peran Penjadwal yang hanya dapat diasumsikan dengan penjadwalan Lambda tertentu dalam 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 EBS Terenkripsi EC2

Saat menjadwalkan EC2 instance yang dilampirkan ke volume EBS yang dienkrpsi oleh AWS KMS, Anda harus memberikan AWS izin kepada Penjadwal Instance untuk menggunakan kunci terkait. AWS KMS 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 EC2 instance yang menggunakan kunci.

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

KMS Ket Arns untuk EC2

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"  
    }  
]  
}
```

Memulai

Panduan ini berisi ikhtisar singkat dan instruksi untuk menyebarkan solusi dengan cepat. Solusi ini menggunakan [AWS CloudFormation templat dan tumpukan](#) untuk mengotomatiskan penerapannya. CloudFormation Template menentukan AWS sumber daya 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 AWS CloudFormation pemetaan, lalu gunakan AWS CloudFormation konsol 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 AWS CloudFormation template di Anda Akun AWS.
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 AWS CloudFormation template di Anda Akun AWS.
2. Masukkan nilai untuk parameter yang diperlukan.

AWS CloudFormation template

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

Anda dapat mengunduh CloudFormation templat untuk solusi ini sebelum menerapkannya.

[View template](#)

instance-scheduler-on-aws.template - Gunakan template ini untuk meluncurkan solusi dan semua komponen terkait. Konfigurasi default menerapkan AWS Lambda fungsi, tabel Amazon DynamoDB, acara CloudWatch Amazon, dan metrik kustom 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, menerapkan 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 `instance-scheduler-on-aws` AWS CloudFormation `template.template`.
2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi di tempat lain Wilayah AWS, 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 [IAM dan AWS STS kuota](#) di AWS Identity and Access Management Panduan Pengguna.
5. Di bawah Parameter, tinjau parameter untuk templat solusi ini dan modifikasi sesuai kebutuhan. Solusi ini menggunakan nilai default berikut.

Parameter	Default
Kunci tag jadwal	Schedule
Interval penjadwalan (menit)	5
Zona waktu default	UTC

Parameter	Default
Penjadwalan diaktifkan	Yes
Aktifkan Penjadwalan xxx	Enabled
Mulai tag	InstanceScheduler-LastA n=Started By {scheduler {year}/{month}/{day} {H inute}{timezone},>
Hentikan tag	InstanceScheduler-LastA n=Stopped By {scheduler {year}/{month}/{day} {H inute}{timezone},>
Aktifkan EC2 jendela pemeliharaan SSM	No
Kunci KMS untuk ARNs EC2	<Optional Input>
Buat snapshot instance RDS saat berhenti	No
Kunci tag terjadwal ASG	scheduled
Awalan nama tindakan ASG	is-

Parameter	Default
Gunakan AWS Organizations	No
Namespace	default
ID Organisasi/Akun Jarak Jauh IDs	<Optional Input>
Wilayah	<Optional Input>
Penjadwalan akun hub yang diaktifkan	Yes
Periode retensi log (hari)	30
Aktifkan CloudWatch Log Debug	No
Pemantauan Operasional	Enabled
Ukuran Memori	128

Parameter	Default
Lindungi Tabel DynamoDB	Enabled

6. Pilih Berikutnya.
7. Pada Konfigurasi halaman opsi stack, 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 pada konsol AWS CloudFormation pada kolom Status. Anda akan menerima status `CREATE_COMPLETE` dalam waktu kurang lebih 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.

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

Launch solution

1. Masuk ke akun sekunder AWS Management Console yang berlaku dan pilih tombol untuk meluncurkan AWS CloudFormation template `instance-scheduler-on-aws-remote`.
2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi di tempat lain Wilayah AWS, gunakan pemilih Wilayah di bilah navigasi konsol. Jika tumpukan hub dikonfigurasi untuk digunakan 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 AWS Organizations opsi 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 Organisasi disetel ke Tidak, maka tumpukan hub harus diperbarui dengan ID Akun baru.

Parameter	Default	Deskripsi
ID Akun Hub	<Requires Input>	ID Akun Penjadwal Instance di tumpukan AWS hub yang akan menjadwalkan sumber daya di akun ini.
Gunakan AWS Organizations	No	Gunakan AWS Organisations untuk mengotomatiskan pendaftaran akun spoke. Harus disetel ke nilai yang sama dengan tumpukan hub.
Namespace	default	Pengidentifikasi unik yang digunakan untuk membedakan antara beberapa penerapan solusi. Harus disetel ke nilai yang sama dengan tumpukan hub.
Kunci Kms untuk ARNs EC2	<Optional Input>	Daftar KMS yang dipisahkan koma untuk memberikan izin solusi ARNs untuk menyediakan EC2 layanan dengan kms: CreateGrant izin dekripsi untuk volume EBS terenkripsi. Ini memungkinkan penjadwal

Parameter	Default	Deskripsi
		<p>untuk memulai EC2 instance dengan volume EBS terenkripsi terlampir. Menyediakan (*) untuk memberikan akses terbatas ke semua kunci KMS; biarkan kosong untuk menonaktifkan. Untuk detail tentang kebijakan persis yang dibuat, lihat Volume EBS Terenkripsi EC2</p>

5. Pilih Berikutnya.
6. Pada halaman Opsi, pilih Selanjutnya.
7. Pada halaman Tinjau dan buat, tinjau dan konfirmasi pengaturan. Pastikan untuk mencentang kotak yang mengakui bahwa template akan membuat sumber daya IAM.
8. Pilih Kirim untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di AWS CloudFormation Konsol 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 [Mengonfigurasi jadwal](#) dan [contoh Tag untuk penjadwalan](#).

Panduan operator

Panduan ini ditujukan untuk pengguna dan operator solusi ini dan berisi detail tentang cara [mengonfigurasi jadwal](#), [memantau solusi](#), [memperbarui solusi](#), dan [fitur-fitur canggih](#) lainnya.

Konfigurasi jadwal

Setelah solusi berhasil diterapkan, Anda dapat mulai mengonfigurasi jadwal. Instance Scheduler on 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)

Instance Scheduler on 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 AWS hub, 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.

Tag instance untuk penjadwalan

Saat menerapkan AWS CloudFormation template, Anda menentukan nama (kunci tag) untuk tag kustom solusi. Untuk Penjadwal Instance aktif AWS untuk mengenali instans Amazon EC2 atau Amazon RDS, kunci tag pada instance tersebut harus cocok dengan kunci tag kustom 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 EC2 sumber daya Amazon Anda](#) dan [Menandai sumber daya Amazon RDS](#).

Pada 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.

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 tersebut hanya dapat berisi rangkaian huruf Unicode, angka, spasi, '_', ':', '/', '=', '+', '-' (Java regex: "`^[\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*$`"). Untuk informasi selengkapnya, lihat [Menandai sumber daya Amazon RDS](#).

EC2 instance 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 KMS: CreateGrant izin untuk dapat memulai instance tersebut. Untuk informasi selengkapnya, lihat Volume [EC2 EBS Terenkripsi](#).

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 Instance on AWS akan menerapkan tindakan awal 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](#).

Bidang hibernasi

Bidang hibernasi memungkinkan Anda menggunakan hibernasi untuk instans Amazon yang dihentikan. EC2 Jika bidang ini disetel ke true, EC2 instance Anda harus menggunakan Amazon

Machine Image (AMI) yang mendukung hibernasi. Untuk informasi selengkapnya, lihat [Linux yang Didukung AMIs dan Windows yang Didukung AMIs](#) di Panduan EC2 Pengguna Amazon. 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 EC2 instance)

`ssm-maintenance-window` Bidang ini memungkinkan Anda untuk secara otomatis menambahkan jendela pemeliharaan AWS Systems Manager sebagai periode berjalan ke jadwal. Saat Anda menentukan nama jendela pemeliharaan yang ada di akun yang sama dan Wilayah AWS sebagai EC2 instans Amazon Anda, solusi akan memulai instance 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 Instance AWS aktif akan memastikan bahwa instans Anda dimulai setidaknya 10 menit sebelum jendela pemeliharaan dimulai. Bergantung pada nilai yang Anda tetapkan untuk AWS CloudFormation parameter 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 keys.

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

Jenis instans

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

Definisi jadwal

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

Bidang	Deskripsi
<code>description</code>	Deskripsi opsional dari jadwal.
<code>hibernate</code>	Pilih apakah akan hibernasi EC2 instans Amazon 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.
<code>enforced</code>	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 dihentikan secara manual selama periode berjalan.
<code>name</code>	Nama yang digunakan untuk mengidentifikasi jadwal. Nama ini harus unik dan hanya mencakup alfa-numerik, tanda hubung (-), dan garis bawah (_).
<code>periods</code>	Nama periode yang digunakan dalam jadwal ini. Masukkan nama persis seperti yang muncul di bidang nama periode. <instance-type>Anda juga dapat menentukan jenis instance untuk periode menggunakan sintaks <code>@<period-name></code> . Misalnya, <code>weekdays@t2.large</code> .

Bidang	Deskripsi
<code>retain_running</code>	Pilih apakah akan mencegah solusi menghentikan instance di akhir periode berjalan jika instance dimulai secara manual sebelum awal periode.
<code>ssm_maintenance_window</code>	Pilih apakah akan menambahkan jendela pemeliharaan AWS Systems Manager sebagai periode berjalan tambahan untuk jadwal ini. Menerima nama jendela StringSet pemeliharaan yang akan dicocokkan dengan nama jendela di akun/wilayah yang sama dengan instance terjadwal. EC2 Catatan: Fitur ini hanya berlaku untuk EC2 instance.
<code>stop_new_instances</code>	Pilih apakah akan menghentikan instance saat pertama kali ditandai jika sedang berjalan di luar periode berjalan. Secara default, bidang ini diatur ke true.
<code>timezone</code>	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 .
<code>use_metrics</code>	Pilih apakah akan mengaktifkan CloudWatch metrik di tingkat jadwal. Bidang ini menimpa setelan CloudWatch metrik yang Anda tentukan saat penerapan. Catatan: Mengaktifkan fitur ini akan dikenakan biaya sebesar \$0,90/bulan per jadwal atau layanan terjadwal.

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 Instance AWS untuk menerapkan tindakan mulai atau berhenti yang sesuai.

Mulai dan hentikan waktu

`endtime` dan `begintime` menentukan kapan Instance Scheduler aktif 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, instance 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 di awal/akhir 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.



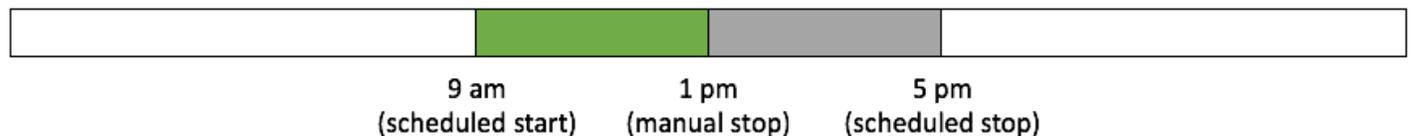
9-5 dijadwalkan mulai dan berhenti

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.



5 PM dijadwalkan berhenti

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 `weekdays`, `monthdays`, or `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 ^{ke-n} 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` Bidang mendefinisikan hari mana selama bulan sebuah instance akan berjalan. Anda dapat menentukan daftar hari, rentang hari, setiap hari ^{ke-n} 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 bulan ^{ke-n}. Solusinya mendukung nama bulan yang disingkat (Jan) dan angka (1).

Definisi periode

Penjadwal Instance pada tabel AWS konfigurasi 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 nth bulan atau setiap nth bulan dalam kisaran. 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
monthdays	<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 hari^{ke-n} dalam sebulan atau setiap hari^{ke-n} dalam satu bulan dalam rentang tertentu. Misalnya, enter 1/7 untuk menjalankan instance setiap hari ketujuh dimulai pada tanggal 1. Masukkan 1-15/2 untuk menjalankan instance setiap hari dari 1st ke 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 tanggal 15.</p>
name	<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 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>

Ketika periode berisi beberapa kondisi, perhatikan bahwa semua kondisi harus benar untuk Penjadwal Instance 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.

Penandaan otomatis

Penjadwal Instance aktif AWS dapat secara otomatis menambahkan tag ke semua instance yang dimulai atau dihentikan. Anda dapat menentukan daftar nama tag atau `tagname=tagvalue` pasangan di tag Mulai dan parameter tag Berhenti. Solusinya juga mencakup makro yang memungkinkan Anda menambahkan informasi variabel ke tag:

- `{scheduler}`: Nama tumpukan penjadwal
- `{year}`: Tahun (empat digit)

- `{month}`: Bulan (dua digit)
- `{day}`: Hari (dua digit)
- `{hour}`: Jam (dua digit, format 24 jam)
- `{minute}`: Menit (dua digit)
- `{timezone}`: Zona waktu

Tabel berikut memberikan contoh input yang berbeda dan tag yang dihasilkan.

Contoh masukan parameter	Tag Penjadwal Instance
<code>ScheduleMessage=Started by scheduler {scheduler}</code>	<code>ScheduleMessage=Started by scheduler MyScheduler</code>
<code>ScheduleMessage=Started on {year}/{month}/{day}</code>	<code>ScheduleMessage=Started on 2017/07/06</code>
<code>ScheduleMessage=Started on {year}/{month}/{day} at {hour}:{minute}</code>	<code>ScheduleMessage=Started on 2017/07/06 at 09:00</code>
<code>ScheduleMessage=Started on {year}/{month}/{day} at {hour}:{minute} {timezone}</code>	<code>ScheduleMessage=Started on 2017/07/06 at 09:00 UTC</code>

Saat Anda menggunakan parameter tag Mulai, tag akan dihapus secara otomatis saat penjadwal menghentikan instance. Saat Anda menggunakan parameter Tag Berhenti, tag akan dihapus secara otomatis saat instance dimulai.

Jadwal sampel

Penjadwal Instance aktif 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 Penjadwal AWS CloudFormation Instance, 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 [CLI Penjadwal Instance](#), 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 AWS CloudFormation konsol dengan memilih Stack [Hub Penjadwal Instance yang digunakan sebelumnya](#) dan kemudian 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 Penjadwal

AWS CloudFormation Instance, 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 AWS CloudFormation konsol dengan mengklik Stack [Hub Penjadwal Instance yang sebelumnya digunakan](#) 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		Nilai
begintime	String	00:00
endtime	String	16:59
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-berhenti-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 Penjadwal AWS CloudFormation Instance, 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 AWS CloudFormation konsol dengan memilih Stack [Hub Penjadwal Instance yang digunakan sebelumnya](#) dan kemudian 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 Instance di AWS tumpukan.

Nama sumber daya	Tipe	Deskripsi
Utama	AWS::Lambda::Function	Penjadwal Instance pada AWS Lambda fungsi.
Penolong Config Scheduler	Custom::ServiceSetup	Menyimpan pengaturan konfigurasi global di Amazon DynamoDB.
Izin Pemanggilan Penjadwal	AWS::Lambda::Permission	Mengizinkan CloudWatch acara Amazon menjalankan fungsi Penjadwal Instance. AWS Lambda
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 EC2 instans Amazon, menetapkan tag, dan mengakses sumber daya penjadwal.

Nama sumber daya	Tipe	Deskripsi
Aturan Penjadwal	<code>AWS::Events::Rule</code>	Aturan EventBridge acara Amazon yang memanggil fungsi Lambda penjadwal.
Aturan Peristiwa Metrik Konfigurasi	<code>AWS::Events::Rule</code>	Aturan EventBridge peristiwa Amazon yang secara berkala memanggil fungsi metrik anonim deskripsi konfigurasi. Dinonaktifkan saat metrik anonim dinonaktifkan.
Tabel Negara	<code>AWS::DynamoDB::Table</code>	Tabel DynamoDB yang menyimpan status instance terakhir yang diinginkan.
Tabel Config	<code>AWS::DynamoDB::Table</code>	Tabel DynamoDB yang menyimpan data konfigurasi, jadwal, dan periode global.
Topik SNS Penjadwal Instance	<code>AWS::SNS::Topic</code>	Mengirim pesan peringatan dan kesalahan ke alamat email berlangganan.

Penjadwal CLI

Penjadwal Instance pada antarmuka baris AWS perintah (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 kredensi untuk AWS CLI. Untuk informasi selengkapnya, lihat [Konfigurasi dan pengaturan file kredensi](#) di Panduan AWS CLI Pengguna.

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 AWS Lambda fungsi dari tumpukan untuk menangani permintaan CLI

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

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
```

5. 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 <i><stackname></i></code>	Nama tumpukan penjadwal. Penting: Argumen ini diperlukan untuk semua perintah.
<code>--region <i><regionname></i></code>	Nama wilayah tempat tumpukan penjadwal digunakan.

Pendapat	Deskripsi
	Catatan: Anda harus menggunakan argumen ini ketika konfigurasi default dan file kredensi tidak diinstal di wilayah yang sama dengan tumpukan solusi.
<code>--profile-name <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 lebih lanjut tentang mengontrol output, lihat Controlling Command Output dari AWS Command Line Interface dalam Panduan AWS CLI Pengguna.
<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

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

- [membantu](#)

menciptakan-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 hari ^{ke-n} 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 bulan ^{ke-n}.

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 hari ^{ke-n} 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>@<instance type>` untuk menentukan jenis instance untuk suatu periode. Misalnya, `weekdays@t2.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 AWS Systems Manager pemeliharaan sebagai periode berjalan ke jadwal EC2 instans Amazon. Untuk menggunakan perintah ini, Anda harus menggunakan `use-maintenance-window` perintah.

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 AWS Systems Manager pemeliharaan sebagai periode berjalan ke jadwal instans Amazon EC2

Wajib: Tidak

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@2.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.

help

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 templat 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 akun/wilayah yang berisi penyebaran hub Anda dan buka konsol. AWS CloudFormation Temukan Stack Hub Penjadwal Instance dan pilih Perbarui -> Gunakan Template yang Ada. Perbarui parameter konfigurasi global yang ingin Anda ubah, lalu pilih berikutnya -> berikutnya -> kirim 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.

Instance Scheduler on AWS menyediakan resource kustom (`ServiceInstanceSchedule`) yang dapat Anda gunakan untuk mengonfigurasi dan mengelola jadwal. 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](#).

Bila 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 `Nama` 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 yang telah selesai, 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 pada. AWSARN 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.

Fitur lanjutan

EC2 Penjadwalan Grup Auto Scaling

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

Lihat [Penskalaan terjadwal untuk EC2 Auto Scaling Amazon 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

Aturan penskalaan terjadwal kemudian akan dikelola untuk ASG Anda oleh dua sistem:

Pertama, fungsi Lambda orkestrator ASG berjalan setiap jam dan memulai fungsi penangan ASG untuk setiap akun/wilayah terjadwal Anda. Fungsi ini akan mencari yang baru ditandai ASGs atau yang tindakan penskalaan terjadwal ASGs yang dikonfigurasi telah menjadi basi. Ini kemudian akan mengkonfigurasi ulang semua tindakan penskalaan terjadwal yang dimulai dengan awalan nama tindakan ASG (ditentukan pada penerapan solusi) agar sesuai dengan jadwal terkait.

Kedua, ketika jadwal diperbarui dalam tabel konfigurasi scheduler, aliran DynamoDB akan memulai (melalui fungsi Schedule Update Handler Lambda) menambahkan permintaan penangan ASG yang akan memperbarui tindakan penskalaan terjadwal pada semua yang ditandai dengan jadwal yang baru diperbarui. ASGs

Definisi Berjalan/Berhenti 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 ASG. min-desired-max

Saat Penjadwal Instance pertama mengonfigurasi Tindakan Penskalaan Terjadwal untuk ASG, min-desired-max nilai yang saat ini dikonfigurasi akan digunakan untuk menentukan status berjalan ASG. Jika ASG saat ini dikonfigurasi dengan 0-0-0, Penjadwal Instance akan melaporkan kesalahan dan tidak akan mengonfigurasi Tindakan Penskalaan Terjadwal hingga baru min-desired-max telah dikonfigurasi yang dapat digunakan untuk menentukan status berjalan untuk ASG. min-desired-max

Saat memperbarui Tindakan Penskalaan Terjadwal untuk ASG, Penjadwal Instance akan melihat min-desired-max saat ini pada saat pembaruan dan menggunakan nilai-nilai tersebut untuk menentukan status berjalan baru untuk jadwal. Jika saat min-desired-max ini 0-0-0 pada saat pembaruan, status berjalan sebelumnya akan digunakan.

Untuk semua ASGs status berhenti didefinisikan sebagai min-desired-max 0-0-0.

Tag Terjadwal ASG

Saat grup penskalaan otomatis dijadwalkan oleh solusi, tag terjadwal grup penskalaan otomatis ditambahkan ke grup penskalaan otomatis. Tag berisi informasi di bawah ini dalam format JSON:

Kunci	Jenis Nilai	Nilai
<code>schedule</code>	String	Nama jadwal selaras dengan tabel konfigurasi scheduler.
<code>ttl</code>	String	Sampai kapan tag valid.
<code>min_size</code>	Bilangan Bulat	Ukuran min grup penskalaan otomatis saat dijadwalkan.
<code>max_size</code>	Bilangan Bulat	Ukuran maksimal grup penskalaan otomatis saat dijadwalkan.
<code>desired_size</code>	Bilangan Bulat	Grup penskalaan otomatis kapasitas yang diinginkan saat dijadwalkan.

Kehadiran Tag Terjadwal yang valid yang belum kedaluwarsa TTL-nya menunjukkan kepada Penjadwal Instance bahwa ASG telah dikonfigurasi dengan benar untuk penjadwalan. Tag ini dapat dihapus secara manual untuk secara paksa menyebabkan Penjadwal Instance mengkonfigurasi ulang Tindakan Penskalaan Terjadwal pada ASG selama penjadwalan ASG berikutnya dijalankan.

Batasan

Penjadwalan ASG dilakukan dengan mengubah Penjadwal Instance pada AWS jadwal 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 Instance AWS secara langsung menerjemahkan beginning/end of periods to start/stop tindakan untuk ASG

bahkan ketika periode lain yang tumpang tindih atau berdekatan biasanya akan menyebabkan tindakan tersebut dilewati.

Pantau solusinya

Pencatatan dan pemberitahuan

Penjadwal Instance di AWS memanfaatkan CloudWatch Log Amazon untuk pencatatan. 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. Untuk informasi selengkapnya, lihat [Sumber daya Solusi](#).

Pesan peringatan dan kesalahan juga dipublikasikan ke topik Amazon SNS yang dibuat solusi, yang mengirim pesan ke alamat email berlangganan. Untuk detailnya, lihat Apa itu Amazon SNS? di Panduan Pengembang Amazon SNS. Anda dapat menemukan nama topik Amazon SNS di tab Output dari tumpukan solusi.

File log

Penjadwal Instance pada AWS membuat grup log yang berisi file AWS Lambda log default dan grup log yang berisi file log berikut:

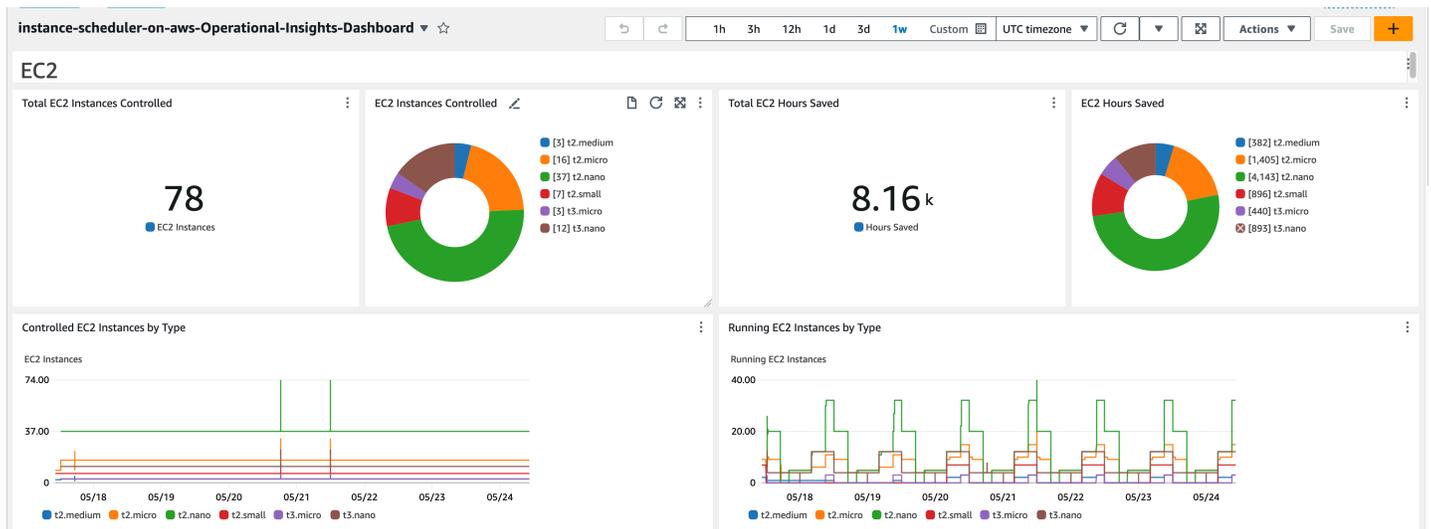
- `InstanceScheduler-yyyyymmdd`: Log pesan penjadwal umum
- `SchedulingOrchestratorHandler-yyyyymmdd`: Log informasi orkestrasi umum saat penjadwalan eksekusi dimulai
- `SchedulerSetupHandler-yyyyymmdd`: Log output dari tindakan konfigurasi
- `Scheduler-<service>-<account>-<region>-yyyyymmdd`: Aktivitas penjadwalan log di setiap layanan, akun, dan wilayah
- `CliHandler-yyyyymmdd`: Permintaan log dari CLI admin
- `Eventbus_request_handler-yyyyymmdd`: Log panggilan ke EventBus sumber daya, jika solusi diterapkan ke AWS organisasi.
- `CollectConfigurationDescription-yyyyymmdd`: Data metrik deskripsi konfigurasi log yang dikirim secara berkala

Dasbor wawasan operasional

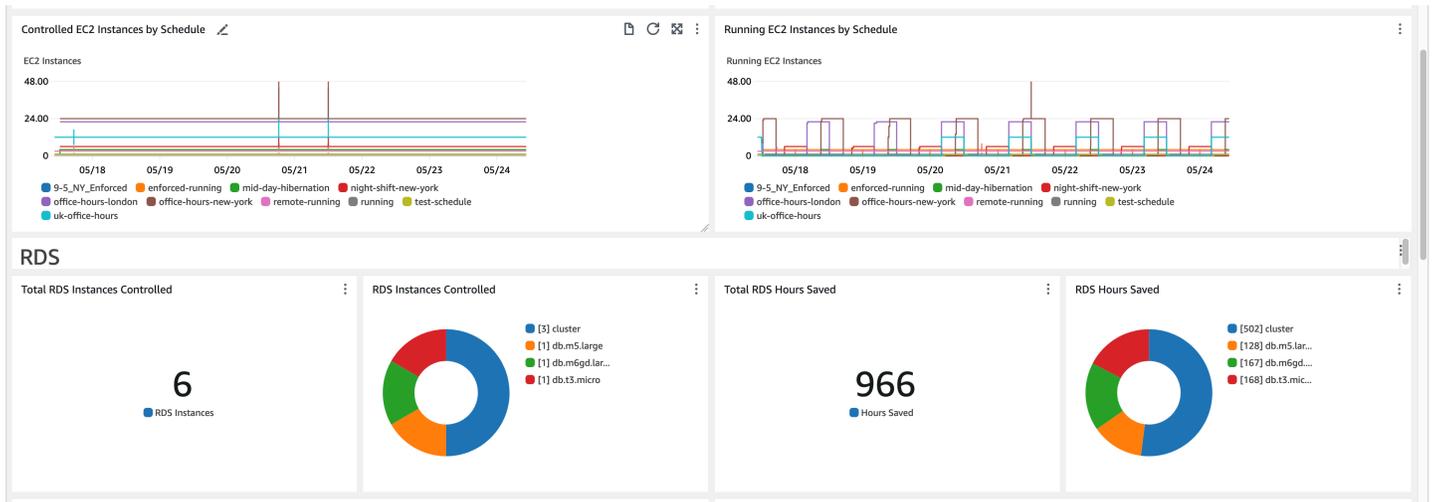
Instance Scheduler on AWS dilengkapi dengan dashboard Operational Insights yang memungkinkan Anda memantau pengoperasian solusi dan mendapatkan wawasan tentang jam berjalan yang telah disimpan dengan menggunakan solusi ini.

Untuk menggunakan dasbor ini, pastikan bahwa Pemantauan Operasional disetel ke “diaktifkan” dalam parameter tumpukan hub solusi di AWS CloudFormation. Lalu pergi ke AWS CloudWatch dan pilih “Dasbor” dari menu navigasi. Nama dasbor akan menjadi {stack-name} -Operational-Insights-Dashboard

Dasbor akan menampilkan berbagai metrik operasional tentang pengoperasian solusi Anda termasuk hitungan berapa banyak instans yang saat ini dikelola oleh solusi, informasi tentang kapan dan berapa banyak instans yang berjalan sepanjang hari, dan perkiraan berapa banyak jam berjalan telah disimpan dengan mematikan instance. Contoh data di bawah ini:



Penjadwal Instance AWS ditumpuk CloudWatch

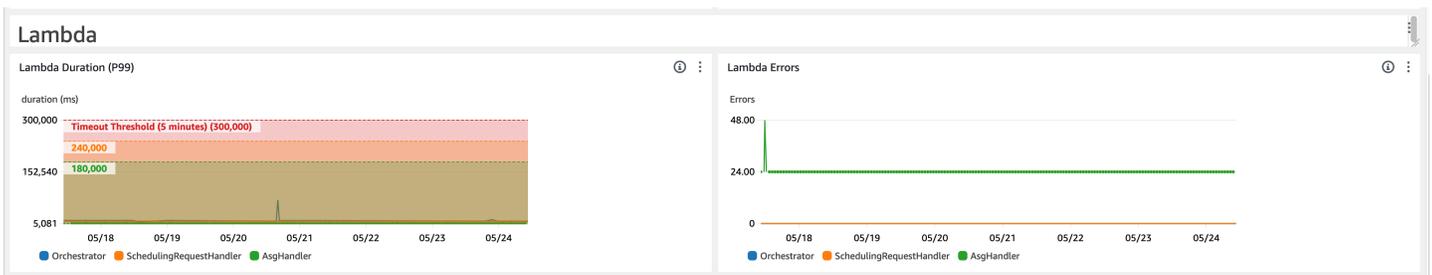


EC2 Instans terkontrol berdasarkan Jadwal

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.

Dasbor juga memberikan wawasan tentang kesehatan fungsi Lambda yang sangat penting untuk pengoperasian solusi. Jika durasi Lambda rata-rata untuk salah satu fungsi Lambda yang digambarkan mulai mendekati zona kuning, mungkin sudah waktunya untuk meningkatkan properti ukuran Lambda pada tumpukan hub solusi.



Durasi Lambda

Digambarkan: AsgHandler meningkatkan kesalahan yang konsisten selama beberapa hari. Ini menunjukkan potensi masalah dengan penjadwalan ASG dan harus meminta penyelidikan lebih lanjut ke dalam log untuk Lambda itu

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
Metrik per jadwal	\$0.60 per jadwal*
Metrik-metrik Per-instance-type	\$0,90 per jenis instance*
Penggunaan API	~ \$0,10 per akun/wilayah

* Biaya ini dilacak per kategori layanan (EC2/RDS) and only for schedules/instance types actually used for scheduling. For example, if you have 15 schedules configured, with three for RDS and five for EC2, the total cost will be $8 * \$0.60$ or \$4.80/month. Jadwal tidak aktif tidak akan ditagih.

Pantau solusinya dengan Service Catalog AppRegistry

Solusinya mencakup AppRegistry sumber daya Service Catalog untuk mendaftarkan CloudFormation template dan sumber daya yang mendasarinya sebagai aplikasi di [Service Catalog AppRegistry](#) dan [AWS Systems Manager Application Manager](#).

AWS Systems Manager Application Manager memberi Anda tampilan tingkat aplikasi ke dalam solusi ini dan sumber dayanya sehingga Anda dapat:

- Pantau sumber dayanya, biaya untuk sumber daya yang digunakan di seluruh tumpukan dan Akun AWS, dan log yang terkait dengan solusi ini dari lokasi pusat.
- Lihat data operasi untuk sumber daya solusi ini dalam konteks aplikasi, seperti status penerapan, CloudWatch alarm, konfigurasi sumber daya, dan masalah operasional.

Gambar berikut menggambarkan contoh tampilan aplikasi untuk Instance Scheduler pada AWS stack di Application Manager.

The screenshot displays the AWS Systems Manager console for an application named 'mpe-v-1-3-0'. The interface includes a left-hand navigation pane with 'Components (2)' and a list of components. The main content area shows 'Application information' with details such as 'Application type: AWS-AppRegistry', 'Name: mpe-v-1-3-0-personalized-experiences-ML', and 'Application monitoring: Enabled'. Below this, there are tabs for 'Overview', 'Resources', 'Compliance', 'Monitoring', 'OpsItems', 'Logs', and 'Runbooks'. The 'Monitoring' tab is active, showing 'Insights and Alarms' with a green 'OK' status for 14 units (100%). It also displays 'Application Insights' with zero problems detected by severity (High, Medium, Low). To the right, the 'Cost' section shows a line graph for 'Cost (USD)' and a table with zero costs for September, October, and November 2022.

Tumpukan solusi di Manajer Aplikasi

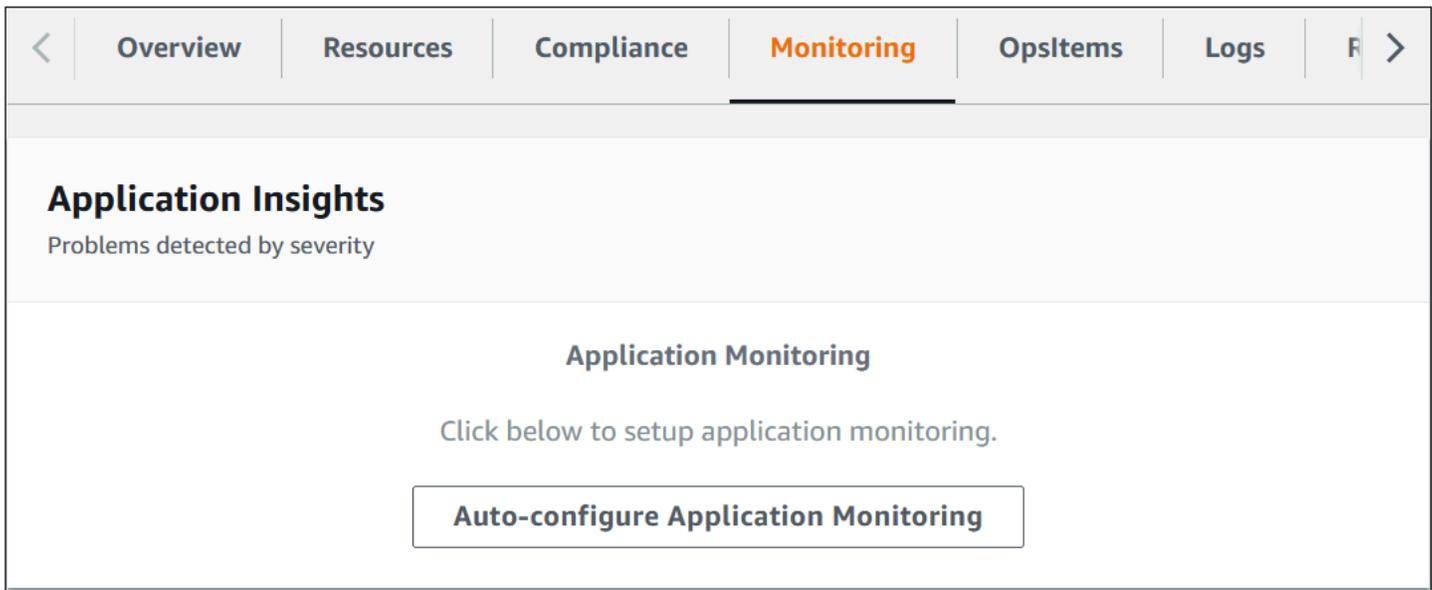
Catatan: Anda harus mengaktifkan CloudWatch Application Insights AWS Cost Explorer, dan tag alokasi biaya yang terkait dengan solusi ini. Mereka tidak diaktifkan secara default.

Aktifkan Wawasan CloudWatch Aplikasi

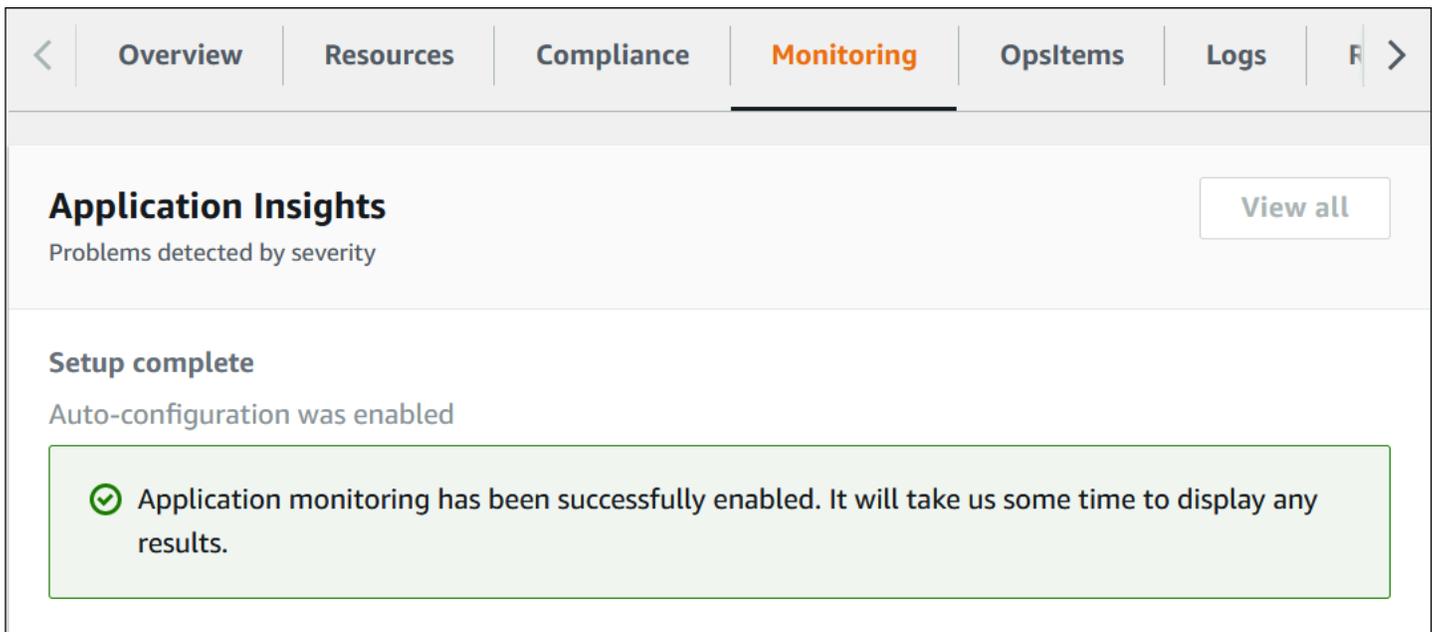
1. Masuk ke [konsol Systems Manager](#).
2. Pada panel navigasi, pilih Manajer Aplikasi.
3. Di Aplikasi, cari nama aplikasi untuk solusi ini dan pilih.

Nama aplikasi akan memiliki App Registry di kolom Sumber Aplikasi, dan akan memiliki kombinasi nama solusi, Wilayah, ID akun, atau nama tumpukan.

4. Di pohon Komponen, pilih tumpukan aplikasi yang ingin Anda aktifkan.
5. Di tab Monitoring, di Application Insights, pilih Konfigurasi Otomatis Wawasan Aplikasi.



Pemantauan untuk aplikasi Anda sekarang diaktifkan dan kotak status berikut muncul:

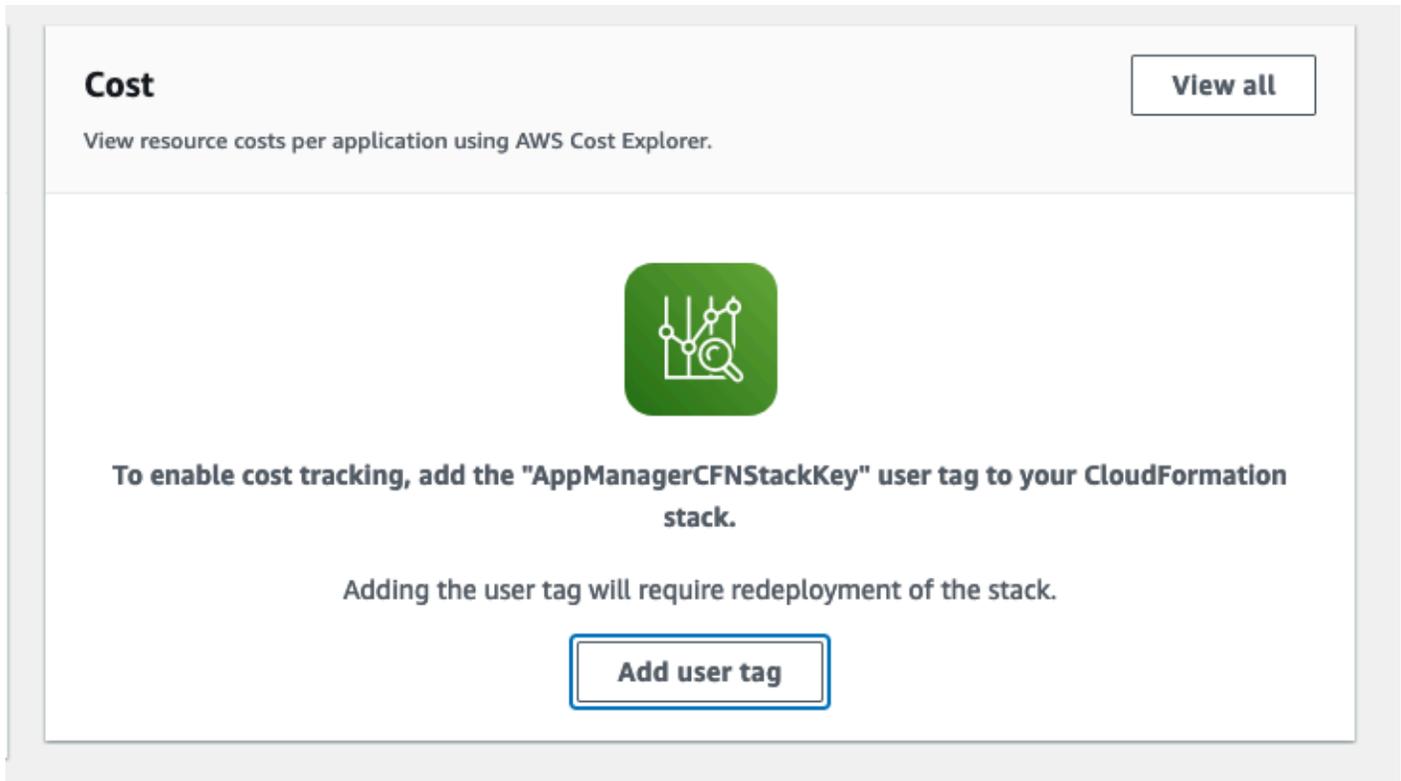


Konfirmasikan tag biaya yang terkait dengan solusi

Setelah Anda mengaktifkan tag alokasi biaya yang terkait dengan solusi, Anda harus mengonfirmasi tag alokasi biaya untuk melihat biaya untuk solusi ini. Untuk mengonfirmasi tag alokasi biaya:

1. Masuk ke [konsol Systems Manager](#).
2. Pada panel navigasi, pilih Manajer Aplikasi.
3. Di Aplikasi, pilih nama aplikasi untuk solusi ini dan pilih.

4. Di tab Ikhtisar, di Biaya, pilih Tambahkan tag pengguna.



5. Pada halaman Tambahkan tag pengguna, masukkan `confirm`, lalu pilih Tambahkan tag pengguna.

Proses aktivasi dapat memakan waktu hingga 24 jam untuk menyelesaikan dan data tag muncul.

Aktifkan tag alokasi biaya yang terkait dengan solusi

Setelah Anda mengaktifkan Cost Explorer, Anda harus mengaktifkan tag alokasi biaya yang terkait dengan solusi ini untuk melihat biaya untuk solusi ini. Tag alokasi biaya hanya dapat diaktifkan dari akun manajemen untuk organisasi. Untuk mengaktifkan tag alokasi biaya:

1. Masuk ke [konsol AWS Manajemen Penagihan dan Biaya dan Manajemen Biaya](#).
2. Di panel navigasi, pilih Tag Alokasi Biaya.
3. Pada halaman Tag alokasi biaya, filter untuk AppManagerCFNStackKey tag, lalu pilih tag dari hasil yang ditampilkan.
4. Pilih Aktifkan.

AWS Cost Explorer

Anda dapat melihat ikhtisar biaya yang terkait dengan komponen aplikasi dan aplikasi dalam konsol Manajer Aplikasi melalui integrasi AWS Cost Explorer yang harus diaktifkan terlebih dahulu. Cost Explorer membantu Anda mengelola biaya dengan memberikan tampilan biaya dan penggunaan AWS sumber daya Anda dari waktu ke waktu. Untuk mengaktifkan Cost Explorer untuk solusinya:

1. Masuk ke [konsol Manajemen AWS Biaya](#).
2. Di panel navigasi, pilih Cost Explorer untuk melihat biaya dan penggunaan solusi dari waktu ke waktu.

Kinerja

Jika AWS Lambda fungsi solusi tidak memproses semua instance terjadwal sebelum pemanggilan berikutnya, solusi akan mencatat kesalahan di Amazon CloudWatch Logs, dan mengirimkan notifikasi Amazon Simple Notification Service (Amazon SNS) ke topik SNS error. Untuk membantu memastikan bahwa semua instance diproses sebelum pemanggilan berikutnya, Anda dapat mengubah interval default di mana fungsi Lambda berjalan atau meluncurkan beberapa penerapan solusi dengan nama tag yang berbeda.

Jika Anda meningkatkan interval default, ini mungkin mengurangi perincian jadwal Anda. Misalnya, fungsi Lambda yang diatur untuk berjalan pada interval 15 menit hanya akan melakukan tindakan mulai dan berhenti setiap 15 menit.

Untuk menjadwalkan sejumlah besar instance, sebaiknya gunakan interval minimal lima menit dan meningkatkan ukuran memori AWS Lambda fungsi utama Instance Scheduler menggunakan parameter Ukuran Memori.

Perbarui solusinya

Important

Instance Scheduler v1.5.0 memiliki masalah kompatibilitas yang diketahui AppRegistry yang mencegahnya untuk dapat langsung meningkatkan ke versi solusi yang lebih baru.

Jika Anda berencana untuk memperbarui dari v1.5.0 ke versi yang AppRegistry diaktifkan di masa mendatang, Anda harus terlebih dahulu memperbarui ke tumpukan perantara 1.5.0-u menggunakan templat berikut:

Tumpukan Hub: <https://solutions-reference.s3.amazonaws.com/aws-instance-scheduler/v1.5.0/aws-instance-scheduler-1.5.0-u.template>

Tumpukan Jarak Jauh: <https://solutions-reference.s3.amazonaws.com/aws-instance-scheduler/v1.5.0/-1.5.0-u.template> [aws-instance-scheduler-remote](#)

Pemasangan tumpukan ini akan menonaktifkan AppRegistry integrasi pada penerapan Anda, memungkinkan versi solusi yang lebih baru untuk membuat ulang asosiasi.

Contoh jalur peningkatan: 1.5.0 -> 1.5.0-u -> 3.0.4

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

1. Masuk ke [AWS CloudFormation konsol](#), di akun/wilayah tempat tumpukan Hub Anda diinstal, pilih **instance-scheduler-on-aws**, dan pilih Perbarui.
2. Pilih Ganti template saat ini.
3. 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.
4. 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 stack scheduler instance](#).
5. Pilih Berikutnya.
6. Pada Konfigurasi halaman opsi stack, pilih Berikutnya.
7. Pada halaman Ulasan, tinjau dan konfirmasi pengaturan. Centang kotak yang mengakui bahwa template akan membuat sumber daya AWS Identity and Access Management (IAM).
8. Pilih Lihat set perubahan dan verifikasi perubahan.
9. Pilih Perbarui tumpukan untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di AWS CloudFormation konsol 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 memutakhirkan langsung dari versi lama ke versi yang lebih baru tanpa kehilangan data atau gangguan pada operasi penjadwalan kecuali jika secara eksplisit disebut di bawah ini. Saat memperbarui versi tertentu sebelumnya, Anda mungkin perlu mengambil tindakan tertentu untuk versi yang Anda lewati. Misalnya, saat memperbarui dari v1.4.1 ke v3.0.2, ikuti instruksi untuk melanggar perubahan di v1.5.0 dan v3.0.0.

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 Gunakan AWS Organizations untuk mengelola pendaftaran spoke ke depan.
 - i. Jika Anda memilih Ya, ganti ID Organisasi/Akun Jarak Jauh IDs dengan ID Organisasi Anda AWS .
 - 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 (tidak boleh diubah dari sebelumnya).

v3.0.0

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.
- Kunci KMS ARNs untuk digunakan dengan volume EBS terenkripsi pada instans EC2 DB sekarang harus disediakan ke tumpukan hub/spoke di akun masing-masing CloudFormation . (Untuk informasi selengkapnya, lihat Volume [EC2 EBS Terenkripsi](#).)
 - Jika Anda menjadwalkan EC2s dengan Volume EBS Terenkripsi, Anda harus menyalin kunci kunci KMS yang digunakan ke parameter tumpukan hub/spoke Anda.
- 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 EC2 pemeliharaan 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.

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, [Kontak Dukungan](#) memberikan instruksi untuk membuka Dukungan kasus untuk solusi ini.

Resolusi masalah yang diketahui

Masalah: Instans tidak dijadwalkan di akun jarak jauh

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: Pembaruan solusi dari versi apa pun v1.3.x ke v1.5.0

Fungsi Lambda tidak berfungsi, misalnya, penjadwalan tidak dilakukan.

Resolusi

1. Pastikan bahwa pembaruan ke CloudFormation tumpukan telah selesai.
2. Buka CloudFormation konsol dan pilih tumpukan solusi.
3. Pilih tab Sumber Daya.
4. Cari Utama di filter Sumber Daya Pencarian.
5. Pilih fungsi Lambda di kolom ID Fisik.
6. Di konsol Lambda pilih Konfigurasi.
7. Pilih Variabel Lingkungan.
8. Pastikan variabel lingkungan berikut tersedia.
 - AKUN
 - CONFIG_TABLE
 - DDB_TABLE_NAME
 - ENABLE_SSM_MAINTENANCE_WINDOWS
 - ISSUES_TOPIC_ARN
 - LOG_GROUP
 - MAINTENANCE_WINDOW_TABLE
 - METRICS_URL
 - SCHEDULER_FREQUENCY
 - SEND_METRICS
 - SOLUTION_ID
 - STACK_ID
 - STACK_NAME
 - MULAI_EC2_BATCH_SIZE
 - STATE_TABLE
 - TAG_NAME
 - JEJAK
 - USER_AGENT

- USER_AGENT_EXTRA
- UUID_KEY

Masalah: EC2 Instans terenkripsi tidak dimulai

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

Resolusi

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

Kontak Dukungan

Jika Anda memiliki [Support AWS Developer](#), [AWS Business Support](#), atau [AWS Enterprise Support](#), Anda dapat menggunakan Support Center untuk mendapatkan bantuan ahli dengan 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 ini secara rinci.
3. Pilih Lampirkan file.
4. Lampirkan informasi yang Dukungan diperlukan 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.

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 AWS solusi dari AWS Management Console atau dengan menggunakan AWS Command Line Interface. Untuk menghapus instalasi solusi, hapus tumpukan hub di AWS CloudFormation 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 AWS Management Console

1. Masuk ke [konsol AWS CloudFormation](#) tersebut.
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 pemasangan, lihat [Apa yang ada AWS Command Line Interface](#) di Panduan AWS CLI Pengguna. Setelah mengonfirmasi bahwa AWS CLI tersedia, jalankan perintah berikut.

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

<installation-stack-name>

Panduan pengembang

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

Kode sumber

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

Penjadwal Instance pada AWS template dihasilkan menggunakan [AWS CDK](#) Lihat file [README.md](#) untuk informasi tambahan.

Referensi

Bagian ini mencakup informasi tentang fitur opsional untuk mengumpulkan metrik unik untuk solusi ini, [kuota](#), petunjuk ke [sumber daya terkait](#), dan [daftar pembangun](#) yang berkontribusi pada solusi ini.

Pengumpulan data anonim

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. Saat dipanggil, informasi berikut dikumpulkan secara berkala dan dikirim ke AWS:

- ID Solusi - Pengidentifikasi AWS Solution.
- Unique ID (UUID) - Pengidentifikasi unik yang dibuat secara acak untuk setiap Penjadwal Instans pada penerapan AWS.
- Stempel waktu - Stempel waktu pengumpulan data.
- Tindakan Penjadwalan - Hitung seberapa sering instance scheduler mengambil tindakan tertentu terhadap instance dan berapa lama waktu yang dibutuhkan untuk melakukan tindakan tersebut.

Contoh data:

```
num_unique_schedules: 4
num_instances_scanned: 23
duration_seconds: 6.7
actions: [
  {
    action: Started
    instanceType: a1.medium
    instances: 8
    service: ec2
  },
  ...
]
```

- Hitungan Instance - Hitungan jumlah instance dan jadwal yang sedang diproses di setiap Wilayah.

Contoh data:

```
service: [ec2]
```

```
regions: [us-east-1]
num_instances: 35
num_schedules: 6
```

- Deskripsi Deployment Deskripsi ikhtisar penyebaran:

Contoh data:

```
services: [ec2, rds]
regions: [us-east-1, us-east-2]
num_accounts: 6
num_schedules: 12
num_cfn_schedules: 3
default_timezone: UTC
schedule_aurora_clusters: True,
create_rds_snapshots: False,
schedule_interval_minutes: 10,
memory_size_mb: 128,
using_organizations: False,
enable_ec2_ssm_maintenance_windows: True,
num_started_tags: 1,
num_stopped_tags: 0,
schedule_flag_counts: {
  stop_new_instances: 10,
  enforced: 3,
  retain_running: 0,
  hibernate: 0,
  override: 0,
  use_ssm_maintenance_window: 2,
  use_metrics: 0,
  non_default_timezone: 4,
},
```

- Penggunaan CLI Seberapa sering setiap fitur CLI Penjadwal digunakan.

Contoh data:

```
command_used: describe-schedule-usage
```

AWS memiliki data yang dikumpulkan melalui survei ini. Pengumpulan data tunduk pada [Kebijakan Privasi AWS](#). Untuk memilih keluar dari fitur ini, selesaikan langkah-langkah berikut sebelum meluncurkan CloudFormation template.

1. Unduh `instance-scheduler-on-aws.template` [AWS CloudFormation template](#) ke hard drive lokal Anda.
2. Buka CloudFormation template dengan editor teks.
3. Ubah bagian pemetaan CloudFormation template dari:

```
"Send": {  
  "AnonymousUsage": {  
    "Data": "Yes"  
  }  
}
```

kepada

```
"Send": {  
  "AnonymousUsage": {  
    "Data": "No"  
  }  
}
```

4. Masuk ke [CloudFormation konsol AWS](#).
5. Pilih Buat tumpukan.
6. Pada halaman Buat tumpukan, Tentukan templat bagian, pilih Unggah file templat.
7. Di bawah Unggah file templat, pilih Pilih file dan pilih templat yang diedit dari drive lokal Anda.
8. Pilih Berikutnya dan ikuti langkah-langkah di bagian [Luncurkan tumpukan](#) panduan ini.

Kuota

Kuota layanan, juga disebut sebagai batas, adalah jumlah maksimum sumber daya layanan atau operasi untuk Anda Akun AWS.

Kuota untuk AWS layanan dalam solusi ini

Pastikan Anda memiliki kuota yang cukup untuk setiap [layanan yang diterapkan dalam solusi ini](#). Untuk informasi selengkapnya, lihat [AWS service quotas](#).

Gunakan tautan berikut untuk membuka halaman untuk layanan itu. Untuk melihat kuota layanan untuk semua AWS layanan dalam dokumentasi tanpa berpindah halaman, lihat informasi di [titik akhir Layanan dan halaman kuota di PDF sebagai](#) gantinya.

AWS CloudFormation kuota

Anda Akun AWS memiliki AWS CloudFormation kuota 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 selengkapya, lihat [AWS CloudFormation kuota](#) di Panduan AWS CloudFormation Pengguna.

Kuota AWS Lambda

Akun Anda memiliki AWS Lambda kuota Eksekusi Serentak 1000, jika solusinya digunakan di akun di mana ada beban kerja lain yang berjalan dan menggunakan Lambda maka Kuota ini harus diatur ke nilai yang sesuai. Nilai ini dapat disesuaikan, untuk informasi lebih lanjut lihat [Panduan AWS Lambda Memulai](#).

Sumber daya terkait

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

Instance Scheduler on AWS menggunakan fungsi Lambda untuk sering mengevaluasi jadwal yang disimpan dalam konfigurasinya dan memeriksa apakah instance berada dalam keadaan 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 Instance AWS saat ini memungkinkan penjadwalan untuk EC2, RDS, dan Aurora Cluster. Resource scheduler hanya menjadwalkan atau memulai dan menghentikan EC2 instance.

Gunakan penjadwal Sumber Daya untuk mengidentifikasi EC2 instance dan memulai/menghentikannya pada waktu tertentu.

Gunakan Penjadwal Instance pada AWS saat akun harus dipindai secara teratur untuk memulai/menghentikan instance.

Tabel mengidentifikasi solusi mana yang lebih baik berdasarkan skenario.

Skenario	Penjadwal Sumber Daya	Penjadwal Instans di AWS
Jadwalkan instans Amazon Neptunus	Tidak	Ya
Jadwalkan instans Amazon DocumentDB	Tidak	Ya
Jadwalkan instance grup Auto Scaling	Tidak	Ya
Jadwalkan EC2 contoh	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

Revisi

Tanggal	Perubahan
Februari 2018	Rilis awal
Juli 2018	Menambahkan klarifikasi pada waktu mulai dan akhir; menambahkan konfigurasi jadwal contoh
Oktober 2018	Menambahkan informasi tentang volume Amazon EBS terenkripsi; menambahkan klarifikasi tentang panjang nama tumpukan dan pembatasan tag Amazon RDS
Mei 2019	Menambahkan informasi tentang instans Amazon Ec2 yang berhibernasi, penjadwalan instans RDS DB yang merupakan bagian dari kluster Amazon Aurora, argumen CLI jadwal baru, jendela pemeliharaan SSM, integrasi Parameter Store, dan klarifikasi tentang jadwal dengan periode berjalan yang berdekatan
Oktober 2019	Menambahkan informasi mengenai Wilayah AWS tempat Penjadwal Instance AWS telah divalidasi
Maret 2020	Perbaiki bug
Juni 2020	Informasi zona waktu yang diklarifikasi untuk waktu mulai dan berhenti aturan periode; untuk informasi tentang pembaruan dan perubahan untuk v1.3.2, lihat file ChangeLog.md di repositori. GitHub
September 2020	AWS Cloud Development Kit (AWS CDK) (AWS CDK) peningkatan konversi dan dokumentasi; untuk informasi lebih lanjut

Tanggal	Perubahan
	tentang perubahan pada v1.3.3, lihat file ChangeLOG.md di repositori. GitHub
Oktober 2020	Diperbarui template di Lampiran D dan memperbarui dokumentasi untuk Hibernate Field.
April 2021	Memperbarui fungsionalitas Jendela Pemeliharaan SSM untuk penjadwalan EC2 misalnya, solusi yang dikonfigurasi untuk bekerja AWS GovCloud, dan perbaikan bug. Untuk informasi lebih lanjut tentang v1.4.0, lihat file ChangeLog.md di repositori. GitHub
Mei 2022	Pembaruan kecil dan perbaikan bug. Untuk informasi lebih lanjut tentang v1.4.1, lihat file ChangeLog.md di repositori. GitHub
Januari 2023	Pembaruan kecil dan perbaikan bug. Untuk informasi lebih lanjut tentang v1.4.2, lihat file ChangeLog.md di repositori. GitHub
Mei 2023	<p>Bendera penjadwalan <code>override_status</code> yang tidak digunakan lagi, Resource Scheduler vs Instance Scheduler, tambahkan fitur untuk AWS Organizations, sederhanakan penjadwalan lintas akun untuk menghilangkan kebutuhan akan peran lintas akun.</p> <p>Untuk informasi lebih lanjut tentang v1.5.0, lihat file ChangeLog.md di repositori. GitHub</p>

Tanggal	Perubahan
Juli 2023	<p>Menambahkan panduan tentang cara mengelola Infrastruktur sebagai Kode, dan memperbarui ke versi 1.5.x. Pemecahan Masalah Tambahan, Fitur dan Manfaat ditambahkan.</p> <p>Untuk informasi lebih lanjut tentang v1.5.1, lihat file ChangeLog.md di repositori. GitHub</p>
Oktober 2023	<p>Pembaruan kecil, penambalan keamanan, dan menambahkan tabel biaya sampel tambahan dalam dokumentasi. Untuk informasi lebih lanjut tentang v1.5.2, lihat file ChangeLog.md di repositori. GitHub</p>
Oktober 2023	<p>Rilis v1.5.3 Patch keamanan. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub</p>
Desember 2023	<p>Pembaruan dokumentasi untuk menyelesaikan perbedaan, menambahkan bagian tindakan ke bagian tersebut. AppRegistry Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub</p>
Februari 2024	<p>Rilis v1.5.4 Patch keamanan. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub</p>
April 2024	<p>Rilis v1.5.5 Patch keamanan. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub</p>

Tanggal	Perubahan
Juni 2024	Rilis v3.0.0: Rilis utama. Menambahkan dukungan untuk Grup EC2 Auto Scaling, Amazon Neptune, dan Amazon DocumentDB. Menambahkan dasbor wawasan operasional. Pembaruan dokumentasi untuk menyertakan panduan Operator dan Pengembang. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub
Juni 2024	v3.0.1: Pembaruan kecil. Instalasi CLI yang dimodifikasi, Uninstall petunjuk solusi dan tautan yang diperbarui. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub
Juli 2024	v3.0.2: Pembaruan kecil. Patch keamanan. Memperbaiki masalah jalur pembaruan untuk jadwal AWS CloudFormation terkelola. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub
Juli 2024	v3.0.3: Patch keamanan. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub
Agustus 2024	v3.0.4: Pembaruan kecil. Instruksi peningkatan telah diperbarui. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub
September 2024	v3.0.5: Pembaruan kecil. Perbaiki bug penjadwalan hari kerja ke-N, dan perbarui ke izin penjadwalan RDS. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub

Tanggal	Perubahan
November 2024	v3.0.6: Pembaruan kecil. Memperbaiki logika percobaan ulang Amazon EC2 dan Amazon RDS dan memperbarui jendela pemeliharaan Amazon RDS untuk memulai 10 menit lebih awal. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub
November 2024	v3.0.7: Pembaruan kecil. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub
Januari 2025	v3.0.8: Pembaruan kecil. Untuk informasi lebih lanjut, lihat file ChangelOG.md di repositori. GitHub

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. AWS produk atau layanan disediakan “sebagaimana adanya” tanpa jaminan, representasi, atau kondisi apa pun, baik tersurat maupun tersirat. AWS tanggung jawab dan kewajiban kepada pelanggannya dikendalikan oleh AWS perjanjian, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

Instance Scheduler on 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.