Panduan Pengguna

EventBridge Penjadwal

EventBridge Penjadwal: Panduan Pengguna

Table of Contents

Apa itu Amazon EventBridge Scheduler?

Amazon EventBridge Scheduler adalah penjadwal tanpa server yang memungkinkan Anda membuat, menjalankan, dan mengelola tugas dari satu layanan terpusat dan terkelola. Sangat skalabel, EventBridge Scheduler memungkinkan Anda menjadwalkan jutaan tugas yang dapat memanggil lebih dari 270 AWS layanan dan lebih dari 6.000 operasi. API Tanpa perlu menyediakan dan mengelola infrastruktur, atau berintegrasi dengan beberapa layanan, EventBridge Scheduler memberi Anda kemampuan untuk memberikan jadwal dalam skala besar dan mengurangi biaya pemeliharaan.

EventBridge Scheduler memberikan tugas Anda dengan andal, dengan mekanisme bawaan yang menyesuaikan jadwal Anda berdasarkan ketersediaan target hilir. Dengan EventBridge Scheduler, Anda dapat membuat jadwal menggunakan ekspresi cron dan rate untuk pola berulang, atau mengonfigurasi pemanggilan satu kali. Anda dapat mengatur jendela waktu fleksibel untuk pengiriman, menentukan batas coba lagi, dan mengatur waktu retensi maksimum untuk pemicu yang gagal.

Topik

Fitur utama dari EventBridge Scheduler

EventBridge Scheduler menawarkan fitur-fitur utama berikut yang dapat Anda gunakan untuk mengonfigurasi target dan menskalakan jadwal Anda.

- Target Templated - EventBridge Scheduler mendukung target template untuk melakukan operasi umum API menggunakan Amazon, SQS Amazon, SNS Lambda, dan. EventBridge Dengan target yang telah ditentukan sebelumnya, Anda dapat mengonfigurasi jadwal Anda dengan cepat menggunakan konsol EventBridge Scheduler, EventBridge SchedulerSDK, atau. AWS CLI
- Target universal - EventBridge Scheduler menyediakan parameter target universal (UTP) yang dapat Anda gunakan untuk membuat pemicu khusus yang menargetkan lebih dari 270 AWS layanan dan lebih dari 6.000 API operasi sesuai jadwal. DenganUTP, Anda dapat mengonfigurasi pemicu yang disesuaikan menggunakan konsol EventBridge Scheduler, EventBridge SchedulerSDK, atau. AWS CLI

- Jendela waktu fleksibel - EventBridge Penjadwal mendukung jendela waktu yang fleksibel, memungkinkan Anda untuk membubarkan jadwal Anda dan meningkatkan keandalan pemicu Anda untuk kasus penggunaan yang tidak memerlukan pemanggilan target terjadwal yang tepat.
- Retries — EventBridge Scheduler menyediakan pengiriman at-least-once acara ke target, yang berarti bahwa setidaknya satu pengiriman berhasil dengan respons dari target. EventBridge Scheduler memungkinkan Anda untuk mengatur jumlah percobaan ulang untuk jadwal Anda untuk tugas yang gagal. EventBridge Scheduler mencoba ulang tugas yang gagal dengan upaya tertunda untuk meningkatkan keandalan jadwal Anda dan memastikan target tersedia.

Mengakses Scheduler EventBridge

Anda dapat menggunakan EventBridge Scheduler melalui EventBridge konsol, EventBridge Scheduler SDK AWS CLI, atau dengan langsung menggunakan Scheduler. EventBridge API

Menyiapkan Amazon EventBridge Scheduler

Sebelum Anda dapat menggunakan EventBridge Scheduler, Anda harus menyelesaikan langkah-langkah berikut.

Topik

Mendaftar untuk AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua AWS layanan dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Buat pengguna IAM.

Untuk membuat pengguna administrator, pilih salah satu opsi berikut.

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
Di Pusat IAM Identitas (Direkomendasikan)	Gunakan kredensi jangka pendek untuk mengakses. AWS Ini sejalan dengan praktik terbaik keamanan. Untuk informasi tentang praktik terbaik, lihat Praktik terbaik keamanan IAM di Panduan IAM Pengguna.	Mengikuti petunjuk di Memulai di Panduan AWS IAM Identity Center Pengguna.	Konfigurasikan akses terprogram dengan Mengonfigurasi AWS CLI yang akan digunakan AWS IAM Identity Center dalam AWS Command Line Interface Panduan Pengguna.
Di IAM (Tidak direkomendasikan)	Gunakan kredensi jangka panjang untuk mengakses. AWS	Mengikuti petunjuk dalam Membuat pengguna IAM admin pertama dan grup pengguna Anda di Panduan IAM Pengguna.	Konfigurasikan akses terprogram dengan Mengelola kunci akses untuk IAM pengguna di Panduan IAM Pengguna.

Gunakan kebijakan terkelola

Pada langkah sebelumnya, Anda mengatur IAM pengguna dengan kredensional untuk mengakses sumber daya Anda AWS . Dalam kebanyakan kasus, untuk menggunakan EventBridge Scheduler dengan aman, kami menyarankan Anda membuat pengguna, grup, atau peran terpisah dengan

hanya izin yang diperlukan untuk menggunakan Scheduler. EventBridge EventBridge Scheduler mendukung kebijakan terkelola berikut untuk kasus penggunaan umum.

- the section called "`AmazonEventBridgeSchedulerFullAccess`"— Memberikan akses penuh ke EventBridge Scheduler menggunakan konsol dan. API
- the section called "`AmazonEventBridgeSchedulerReadOnlyAccess`"— Memberikan akses hanya-baca ke Scheduler. EventBridge

Anda dapat melampirkan kebijakan terkelola ini ke IAM kepala sekolah Anda dengan cara yang sama seperti Anda melampirkan `AdministratorAccess` kebijakan pada langkah sebelumnya. Untuk informasi selengkapnya tentang mengelola akses ke EventBridge Scheduler menggunakan kebijakan berbasis identitasIAM, lihat. the section called "Menggunakan kebijakan berbasis identitas"

Mengatur peran eksekusi

Peran eksekusi adalah IAM peran yang diasumsikan oleh EventBridge Scheduler untuk berinteraksi dengan orang lain AWS layanan atas nama Anda. Anda melampirkan kebijakan izin ke peran ini untuk memberikan EventBridge Scheduler akses ke target pemanggilan.

Anda juga dapat membuat peran eksekusi baru saat menggunakan konsol untuk membuat jadwal baru. Jika Anda menggunakan konsol, EventBridge Scheduler membuat peran atas nama Anda dengan izin berdasarkan target yang Anda pilih. Saat EventBridge Scheduler membuat peran untuk Anda, kebijakan kepercayaan peran tersebut mencakup kunci kondisi yang membatasi prinsipal mana yang dapat mengambil peran atas nama Anda. Ini menjaga terhadap potensi masalah keamanan wakil yang membingungkan.

Langkah-langkah berikut menjelaskan cara membuat peran eksekusi baru dan cara memberikan EventBridge Scheduler akses untuk memanggil target. Topik ini menjelaskan izin untuk target template populer. Untuk informasi tentang menambahkan izin untuk target lain, lihatthe section called "Menggunakan target template".

Untuk membuat peran eksekusi menggunakan AWS CLI

1. Salin JSON kebijakan peran asumsi berikut dan simpan secara lokal sebagai`Scheduler-Execution-Role.json`. Kebijakan kepercayaan ini memungkinkan EventBridge Scheduler untuk mengambil peran atas nama Anda.

```
{
```

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

> **Important**
>
> T mengatur peran eksekusi dalam lingkungan produksi, kami sarankan menerapkan perlindungan tambahan untuk mencegah masalah wakil yang membingungkan. Untuk informasi selengkapnya dan kebijakan contoh, lihatthe section called “Pencegahan Deputi Bingung”.

2. Dari AWS Command Line Interface (AWS CLI), masukkan perintah berikut untuk membuat peran baru. Ganti *SchedulerExecutionRole* dengan nama yang ingin Anda berikan peran ini.

```
$ aws iam create-role --role-name SchedulerExecutionRole --assume-role-policy-
document file://Scheduler-Execution-Role.json
```

Jika berhasil, Anda akan melihat output berikut:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Scheduler-Execution-Role",
        "RoleId": "BR1L2DZK3K4CTL5ZF9EIL",
        "Arn": "arn:aws:iam::123456789012:role/SchedulerExecutionRole",
        "CreateDate": "2022-03-10T18:45:01+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "scheduler.amazonaws.com"
```

```
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

3. Untuk membuat kebijakan baru yang memungkinkan EventBridge Scheduler memanggil target, pilih salah satu target umum berikut. Salin kebijakan JSON izin dan simpan secara lokal sebagai `.json` file.

   Amazon SQS – SendMessage

   Berikut ini memungkinkan EventBridge Scheduler untuk memanggil `sqs:SendMessage` tindakan di semua SQS antrian Amazon di akun Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:SendMessage"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

   Amazon SNS – Publish

   Berikut ini memungkinkan EventBridge Scheduler untuk memanggil `sns:Publish` tindakan pada semua SNS topik Amazon di akun Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sns:Publish"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Lambda – Invoke

Berikut ini memungkinkan EventBridge Scheduler untuk memanggil `lambda:InvokeFunction` tindakan pada semua fungsi Lambda di akun Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

4. Jalankan perintah berikut untuk membuat kebijakan izin baru. Ganti *`PolicyName`* dengan nama yang ingin Anda berikan pada kebijakan ini.

```
$ aws iam create-policy --policy-name PolicyName --policy-document file:// PermissionPolicy.json
```

Jika berhasil, Anda akan melihat output berikut. Perhatikan kebijakannyaARN. Anda menggunakan ini ARN di langkah berikutnya untuk melampirkan kebijakan ke peran eksekusi kami.

```
{
    "Policy": {
        "PolicyName": "PolicyName",
        "CreateDate": "2022-03-015T19:31:18.620Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
```

```
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:policy/PolicyName",
        "UpdateDate": "2022-03-015T19:31:18.620Z"
    }
}
```

5. Jalankan perintah berikut untuk melampirkan kebijakan ke peran eksekusi Anda. Ganti *your-policy-arn* dengan kebijakan yang Anda buat di langkah sebelumnya. ARN Ganti *SchedulerExecutionRole* dengan nama peran eksekusi Anda.

```
$ aws iam attach-role-policy --policy-arn your-policy-arn --role-
name SchedulerExecutionRole
```

`attach-role-policy`Operasi tidak mengembalikan respons pada baris perintah.

Siapkan target

Sebelum Anda membuat jadwal EventBridge Scheduler, Anda memerlukan setidaknya satu target agar jadwal Anda dapat dipanggil. Anda dapat menggunakan AWS sumber daya yang ada, atau membuat yang baru. Langkah-langkah berikut menunjukkan cara membuat SQS antrian Amazon standar baru dengan AWS CloudFormation.

Untuk membuat SQS antrian Amazon baru

1. Salin JSON AWS CloudFormation template berikut dan simpan secara lokal sebagai`SchedulerTargetSQS.json`.

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "MyQueue": {
         "Type": "AWS::SQS::Queue",
         "Properties": {
            "QueueName": "MyQueue"
             }
         }
      },
   "Outputs": {
      "QueueName": {
```

```
            "Description": "The name of the queue",
            "Value": {
               "Fn::GetAtt": [
                  "MyQueue",
                  "QueueName"
               ]
            }
         },
         "QueueURL": {
            "Description": "The URL of the queue",
            "Value": {
               "Ref": "MyQueue"
            }
         },
         "QueueARN": {
            "Description": "The ARN of the queue",
            "Value": {
               "Fn::GetAtt": [
                  "MyQueue",
                  "Arn"
               ]
            }
         }
      }
}
```

2. Dari AWS CLI, jalankan perintah berikut untuk membuat AWS CloudFormation tumpukan dari `Scheduler-Target-SQS.json` template.

```
$ aws cloudformation create-stack --stack-name Scheduler-Target-SQS --template-body
 file://Scheduler-Target-SQS.json
```

Jika berhasil, Anda akan melihat output berikut:

```
{
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/Scheduler-
Target-SQS/1d2af345-a121-12eb-abc1-012e34567890"
}
```

3. Jalankan perintah berikut untuk melihat informasi ringkasan untuk AWS CloudFormation tumpukan Anda. Informasi ini mencakup status tumpukan dan output yang ditentukan dalam template.

```
$ aws cloudformation describe-stacks --stack-name Scheduler-Target-SQS
```

Jika berhasil, perintah akan membuat SQS antrian Amazon dan mengembalikan output berikut:

```
{
    "Stacks": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
Scheduler-Target-SQS/1d2af345-a121-12eb-abc1-012e34567890",
            "StackName": "Scheduler-Target-SQS",
            "CreationTime": "2022-03-17T16:21:29.442000+00:00",
            "RollbackConfiguration": {},
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false,
            "NotificationARNs": [],
            "Outputs": [
                {
                    "OutputKey": "QueueName",
                    "OutputValue": "MyQueue",
                    "Description": "The name of the queue"
                },
                {
                    "OutputKey": "QueueARN",
                    "OutputValue": "arn:aws:sqs:us-west-2:123456789012:MyQueue",
                    "Description": "The ARN of the queue"
                },
                {
                    "OutputKey": "QueueURL",
                    "OutputValue": "https://sqs.us-
west-2.amazonaws.com/123456789012/MyQueue",
                    "Description": "The URL of the queue"
                }
            ],
            "Tags": [],
            "EnableTerminationProtection": false,
            "DriftInformation": {
                "StackDriftStatus": "NOT_CHECKED"
            }
        }
    ]
}
```

Kemudian dalam panduan ini, Anda akan menggunakan nilai `QueueARN` untuk mengatur antrian sebagai target untuk EventBridge Scheduler.

Apa selanjutnya?

Setelah Anda menyelesaikan langkah penyiapan, gunakan panduan Memulai untuk membuat EventBridge penjadwal Penjadwal pertama Anda dan memanggil target.

Memulai dengan EventBridge Scheduler

Topik ini menjelaskan pembuatan EventBridge jadwal Scheduler baru. Anda menggunakan konsol EventBridge Scheduler, AWS Command Line Interface (AWS CLI), atau AWS SDKs untuk membuat jadwal dengan target Amazon SQS template. Kemudian, Anda akan mengatur logging, mengonfigurasi percobaan ulang, dan menetapkan waktu retensi maksimum untuk tugas yang gagal. Setelah membuat jadwal, Anda akan memverifikasi bahwa jadwal Anda berhasil memanggil target dan mengirim pesan ke antrean target.

> **Note**
>
> Untuk mengikuti panduan ini, kami sarankan Anda mengatur IAM pengguna dengan izin minimum yang diperlukan yang dijelaskan dalamthe section called "Menggunakan kebijakan berbasis identitas". Setelah Anda membuat dan mengkonfigurasi pengguna, jalankan perintah berikut untuk mengatur kredensi akses Anda. Anda akan memerlukan ID kunci akses dan kunci akses rahasia untuk mengonfigurasi file AWS CLI.
>
> ```
> $ aws configure
> AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
> AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
> Default region name [None]: us-west-2
> Default output format [None]: json
> ```
>
> Untuk informasi selengkapnya tentang berbagai cara mengatur kredensialnya, lihat Pengaturan dan prioritas konfigurasi di Panduan AWS Command Line Interface Pengguna untuk Versi 2.

Topik

Prasyarat

Sebelum mencoba langkah-langkah di bagian ini, Anda harus melakukan hal berikut:

- Selesaikan tugas yang dijelaskan di Pengaturan

Buat jadwal menggunakan konsol EventBridge Scheduler

Untuk membuat jadwal baru menggunakan konsol

1. Masuk ke AWS Management Console, lalu pilih tautan berikut untuk membuka bagian EventBridge Penjadwal EventBridge konsol: https://us-west-2.console.aws.amazon.com/scheduler/rumah? wilayah=us-barat-2 #home

   > **Note**
   >
   > Anda dapat beralih Wilayah AWS dengan menggunakan pemilih Wilayah. AWS Management Console

2. Pada halaman Jadwal, pilih Buat jadwal.
3. Pada halaman Tentukan detail jadwal, di bagian Nama jadwal dan deskripsi, lakukan hal berikut:

   a. Untuk nama Jadwal, masukkan nama untuk jadwal Anda. Sebagai contoh, **`MyTestSchedule`**.

   b. Untuk Deskripsi - opsional, masukkan deskripsi untuk jadwal Anda. Misalnya, **`My first schedule`**.

   c. Untuk grup Jadwal, pilih grup jadwal dari opsi drop-down. Jika sebelumnya Anda belum membuat grup jadwal, Anda dapat memilih `default` grup untuk jadwal Anda. Untuk membuat grup jadwal baru, pilih tautan buat jadwal Anda sendiri di deskripsi konsol. Anda menggunakan grup jadwal untuk menambahkan tag ke grup jadwal.

4. Di bagian Pola Jadwal, lakukan hal berikut:

   a. Untuk Kejadian, pilih salah satu opsi pola berikut. Opsi konfigurasi berubah tergantung pada pola mana yang Anda pilih.

      - Jadwal satu kali — Jadwal satu kali memanggil target hanya sekali pada tanggal dan waktu yang Anda tentukan.

Untuk Tanggal dan waktu, masukkan tanggal yang valid dalam `YYYY/MM/DD` format. Kemudian, tentukan stempel waktu dalam format 24 jam`hh:mm`. Terakhir, pilih zona waktu dari opsi drop-down.

- Jadwal berulang — Jadwal berulang memanggil target pada tingkat yang Anda tentukan menggunakan cron ekspresi atau ekspresi tingkat.

  Pilih jadwal berbasis Cron untuk mengonfigurasi jadwal dengan menggunakan ekspresi. cron Untuk menggunakan ekspresi tingkat, pilih Jadwal berbasis tarif dan masukkan angka positif untuk Nilai, lalu pilih Unit dari opsi drop-down.

Untuk informasi selengkapnya tentang penggunaan ekspresi cron dan rate, lihatJenis jadwal.

b. Untuk jendela waktu Fleksibel, pilih Nonaktif untuk mematikan opsi, atau pilih salah satu jendela waktu yang telah ditentukan sebelumnya dari daftar drop-down. Misalnya, jika Anda memilih 15 menit dan Anda menetapkan jadwal berulang untuk memanggil targetnya setiap jam sekali, jadwal berjalan dalam 15 menit setelah dimulainya setiap jam.

5. Jika Anda memilih Jadwal berulang pada langkah sebelumnya, di bagian Jangka waktu, tentukan zona waktu, dan secara opsional tetapkan tanggal dan waktu mulai, serta tanggal dan waktu akhir untuk jadwal tersebut. Jadwal berulang tanpa tanggal mulai akan dimulai segera setelah dibuat dan tersedia. Jadwal berulang tanpa tanggal akhir akan terus memanggil targetnya tanpa batas waktu.
6. Pilih Berikutnya.
7. Pada halaman Pilih target, lakukan hal berikut:

   a. Pilih target Templated dan pilih targetAPI. Untuk contoh ini, kita akan memilih target SQS `SendMessage` template Amazon.

   b. Pada SendMessagebagian, untuk SQSantrian, pilih SQS antrian Amazon yang ada ARN seperti `arn:aws:sqs:`*`us-west-2`*`:`*`123456789012`*`:`*`TestQueue`* dari daftar drop-down. Untuk membuat antrean baru, pilih Buat SQS antrean baru untuk menavigasi ke konsol AmazonSQS. Setelah selesai membuat antrian, kembali ke konsol EventBridge Scheduler dan segarkan drop-down. Antrian baru Anda ARN muncul dan dapat dipilih.

   c. Untuk Target, masukkan payload yang ingin Anda kirimkan EventBridge Scheduler ke target. Untuk contoh ini, kami akan mengirim pesan berikut ke antrian target: **`Hello, it's EventBridge Scheduler.`**

8. Pilih Berikutnya, lalu pada halaman Pengaturan - opsional, lakukan hal berikut:
9.
   a. Di bagian Status jadwal, untuk Aktifkan jadwal, aktifkan atau nonaktifkan fitur menggunakan sakelar. Secara default, EventBridge Scheduler memungkinkan jadwal Anda.
   b. Di bagian Tindakan setelah jadwal selesai, konfigurasikan tindakan yang dilakukan EventBridge Penjadwal setelah jadwal selesai:
      - Pilih DELETEapakah Anda ingin jadwal dihapus secara otomatis. Untuk jadwal satu kali, ini terjadi setelah jadwal memanggil target sekali. Untuk jadwal berulang, ini terjadi setelah pemanggilan terakhir jadwal yang direncanakan. Untuk informasi selengkapnya tentang penghapusan otomatis, lihat. the section called “Penghapusan setelah jadwal selesai”
      - Pilih NONE, atau jangan pilih nilai, jika Anda tidak ingin EventBridge Scheduler mengambil tindakan apa pun setelah jadwal selesai.
   c. Di bagian Coba lagi kebijakan dan antrean huruf mati (DLQ), untuk kebijakan Coba lagi, aktifkan Coba lagi untuk mengonfigurasi kebijakan coba lagi untuk jadwal Anda. Dengan kebijakan coba lagi, jika jadwal gagal memanggil targetnya, EventBridge Scheduler menjalankan kembali jadwal. Jika dikonfigurasi, Anda harus mengatur waktu retensi maksimum dan mencoba ulang untuk jadwal.
   d. Untuk Usia maksimum acara - opsional, masukkan jam maksimum dan min yang harus disimpan oleh EventBridge Scheduler untuk menyimpan acara yang belum diproses.

      > **Note**
      > Nilai maksimum adalah 24 jam.

   e. Untuk percobaan ulang Maksimum, masukkan jumlah maksimum kali EventBridge Scheduler mencoba ulang jadwal jika target mengembalikan kesalahan.

      > **Note**
      > Nilai maksimum adalah 185 percobaan ulang.

   f. Untuk antrian Dead-letter (DLQ), pilih dari opsi berikut:
      - Tidak ada — Pilih opsi ini jika Anda tidak ingin mengkonfigurasi fileDLQ.

- Pilih SQS antrian Amazon di AWS akun saya sebagai DLQ — Pilih opsi ini, lalu pilih antrian ARN dari daftar drop-down, DLQ konfigurasikan Akun AWS sama dengan yang Anda buat jadwal.
- Tentukan SQS antrian Amazon di AWS akun lain sebagai DLQ — Pilih opsi ini, lalu masukkan antrean konfigurasi sebagaiDLQ, jika antrean ada di lain. ARN Akun AWS Anda harus memasukkan persis ARN antrian untuk menggunakan opsi ini.

g. Di bagian Enkripsi, pilih Sesuaikan pengaturan enkripsi (lanjutan) untuk menggunakan KMS kunci yang dikelola pelanggan untuk mengenkripsi input target Anda. Jika Anda memilih opsi ini, masukkan KMS kunci yang ada ARN atau pilih Buat AWS KMS kunci untuk menavigasi ke AWS KMS konsol. Untuk informasi selengkapnya tentang cara EventBridge Scheduler mengenkripsi data Anda saat istirahat, lihat. the section called “Enkripsi diam”

h. Untuk Izin, pilih Gunakan peran yang ada, lalu pilih peran yang Anda buat selama prosedur penyiapan dari daftar drop-down. Anda juga dapat memilih Pergi ke IAM konsol untuk membuat peran baru.

   Jika Anda ingin EventBridge Scheduler membuat peran eksekusi baru untuk Anda, pilih Buat peran baru untuk jadwal ini. Kemudian, masukkan nama untuk nama Peran. Jika Anda memilih opsi ini, EventBridge Scheduler menambahkan izin yang diperlukan untuk target template Anda ke peran.

10. Pilih Berikutnya.
11. Di halaman Tinjau dan buat jadwal, tinjau detail jadwal Anda. Di setiap bagian, pilih Edit untuk kembali ke langkah itu dan mengedit detailnya.
12. Pilih Buat jadwal untuk menyelesaikan pembuatan jadwal baru Anda. Anda dapat melihat daftar jadwal baru dan yang sudah ada di halaman Jadwal. Di bawah kolom Status, verifikasi bahwa jadwal baru Anda Diaktifkan.
13. Untuk memverifikasi bahwa jadwal Anda memanggil SQS target Amazon, buka SQS konsol Amazon dan lakukan hal berikut:

    a. Pilih antrian target dari daftar Antrian.

    b. Pilih Kirim dan terima pesan.

    c. Pada halaman Kirim dan terima pesan, di bawah Menerima pesan, pilih Poll untuk pesan untuk mengambil pesan uji yang dikirim oleh jadwal Anda ke antrean target.

Buat jadwal menggunakan AWS CLI

Contoh berikut menunjukkan cara menggunakan AWS CLI perintah `create-schedule`untuk membuat jadwal EventBridge Scheduler dengan target Amazon SQS template. Ganti nilai placeholder untuk parameter berikut dengan informasi Anda:

- --name — Masukkan nama untuk jadwal.
- RoleArn— Masukkan ARN peran eksekusi yang ingin Anda kaitkan dengan jadwal.
- Arn — Masukkan ARN target. Dalam hal ini, targetnya adalah SQS antrian Amazon.
- Input — Masukkan pesan yang EventBridge Scheduler kirimkan ke antrian target.

```
$ aws scheduler create-schedule --name sqs-templated-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Buat jadwal menggunakan EventBridge Scheduler SDKs

Dalam contoh berikut, Anda menggunakan EventBridge Scheduler SDKs untuk membuat jadwal EventBridge Scheduler dengan target Amazon SQS template.

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "Message for scheduleArn: '<aws.scheduler.schedule-arn>', scheduledTime:
 '<aws.scheduler.scheduled-time>'"
}

scheduler.create_schedule(
    Name="sqs-python-templated",
    ScheduleExpression="rate(5 minutes)",
```

```
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

Example Jawa SDK

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target sqsTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<QUEUE_ARN>")
                .input("Message for scheduleArn: '<aws.scheduler.schedule-arn>',
 scheduledTime: '<aws.scheduler.scheduled-time>'")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(sqsTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and an Amazon SQS
 templated target");
    }
}
```

Apa selanjutnya?

- Untuk informasi selengkapnya tentang mengelola jadwal menggunakan konsol AWS CLI, atau EventBridge PenjadwalSDK, lihatMengelola jadwal.
- Untuk informasi selengkapnya tentang cara mengonfigurasi target template dan mempelajari cara menggunakan parameter target universal, lihatMengelola target.
- Untuk informasi selengkapnya tentang tipe dan API operasi data EventBridge Scheduler, lihat Referensi EventBridge Penjadwal API.

Jenis jadwal di EventBridge Scheduler

Topik berikut menjelaskan berbagai jenis jadwal yang didukung Amazon EventBridge Scheduler, serta cara EventBridge Scheduler menangani waktu musim panas, dan penjadwalan di zona waktu yang berbeda. Anda dapat memilih dari tiga jenis jadwal saat mengonfigurasi jadwal Anda: jadwal berbasis tarif, berbasis cron, dan satu kali.

Baik jadwal berbasis tarif dan cron adalah jadwal berulang. Anda mengonfigurasi setiap jenis jadwal berulang menggunakan ekspresi jadwal untuk jenis jadwal yang ingin Anda konfigurasi, dan menentukan zona waktu di mana EventBridge Scheduler mengevaluasi ekspresi.

Jadwal satu kali adalah jadwal yang memanggil target hanya sekali. Anda mengonfigurasi jadwal satu kali dengan menentukan waktu, tanggal, dan zona waktu di mana EventBridge Penjadwal mengevaluasi jadwal.

> **Note**
>
> Semua jenis jadwal pada EventBridge Scheduler memanggil target mereka dengan presisi 60 detik. Ini berarti bahwa jika Anda mengatur jadwal Anda untuk dijalankan`1:00`, itu akan memanggil target API antara `1:00:00` dan`1:00:59`, dengan asumsi bahwa jendela waktu yang fleksibel tidak ditetapkan.

Gunakan bagian berikut untuk mempelajari tentang mengonfigurasi ekspresi jadwal untuk setiap jenis jadwal berulang, dan cara mengatur jadwal satu kali di Scheduler. EventBridge

Topik

Jadwal berbasis tarif

Jadwal berbasis tarif dimulai setelah tanggal mulai yang Anda tentukan untuk jadwal Anda, dan berjalan pada tingkat reguler yang Anda tentukan hingga tanggal akhir jadwal. Anda dapat mengatur kasus penggunaan penjadwalan berulang yang paling umum menggunakan jadwal berbasis tarif. Misalnya, jika Anda menginginkan jadwal yang memanggil targetnya setiap 15 menit, setiap dua jam sekali, atau setiap lima hari sekali, Anda dapat menggunakan jadwal berbasis tarif untuk mencapai ini. Anda mengonfigurasi jadwal berbasis laju menggunakan ekspresi laju.

Dengan jadwal berbasis tarif, Anda menggunakan `StartDate`properti untuk mengatur kemunculan pertama jadwal. Jika Anda tidak `StartDate` menyediakan jadwal berdasarkan tarif, jadwal Anda mulai memanggil target segera.

Ekspresi tingkat memiliki dua bidang wajib yang dipisahkan oleh spasi putih, seperti yang ditunjukkan pada berikut ini.

Sintaks

```
rate(value unit)
```

nilai

Bilangan positif

unit

Unit waktu yang Anda inginkan jadwal Anda untuk memanggil targetnya.

Masukan yang valid: `minutes` | | `hours` `days`

Contoh

Contoh berikut menunjukkan cara menggunakan ekspresi tingkat dengan AWS CLI `create-schedule` perintah untuk mengkonfigurasi jadwal berbasis laju. Contoh ini membuat jadwal yang berjalan setiap lima menit dan mengirimkan pesan ke SQS antrian Amazon, menggunakan tipe target template`SqsParameters`.

Karena contoh ini tidak menetapkan nilai untuk `--start-date` parameter, jadwal mulai memanggil targetnya segera setelah Anda membuat dan mengaktifkannya.

```
$ aws scheduler create-schedule --schedule-expression 'rate(5 minutes)' --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Jadwal berbasis cron

Ekspresi cron menciptakan jadwal berulang berbutir halus yang berjalan pada waktu tertentu yang Anda pilih. EventBridge Scheduler mendukung konfigurasi jadwal berbasis cron di Universal Coordinated Time (UTC), atau di zona waktu yang Anda tentukan saat Anda membuat jadwal. Dengan jadwal berbasis cron, Anda memiliki kontrol lebih besar atas kapan dan seberapa sering jadwal Anda berjalan. Gunakan jadwal berbasis cron saat Anda membutuhkan jadwal pengulangan yang disesuaikan yang tidak didukung oleh salah satu ekspresi tingkat EventBridge Scheduler. Misalnya, Anda dapat membuat jadwal berbasis cron yang berjalan pada pukul 8:00 pagi. PSTpada hari Senin pertama setiap bulan. Anda mengonfigurasi jadwal berbasis cron menggunakan ekspresi cron.

Ekspresi cron terdiri dari lima bidang wajib yang dipisahkan oleh spasi putih: menit, jam, day-of-month, bulan day-of-week, dan satu bidang opsional, tahun, seperti yang ditunjukkan pada berikut ini.

Sintaks

```
cron(minutes hours day-of-month month day-of-week year)
```

Bidang	Nilai-nilai	Wildcard
Menit	0-59	, - * /
Jam	0-23	, - * /
D ay-of-month	1-31	, - * ? / L W
Bulan	1-12 atau JAN - DEC	, - * /
D ay-of-week	1-7 atau SUN - SAT	, - * ? L #
Tahun	1970-2199	, - * /

Wildcard

- Wildcard , (koma) mencakup nilai tambahan. Di bidang Bulan,, JANFEB, MAR termasuk Januari, Februari, dan Maret.
- Wildcard - (tanda hubung) menentukan rentang. Di bidang Tanggal, 1-15 mencakup tanggal 1 hingga 15 pada bulan yang ditentukan.
- Wildcard * (bintang) mencakup semua nilai di bidang. Di bidang Jam, * mencakup setiap jam. Anda tidak dapat menggunakan* di ay-of-week bidang D ay-of-month dan D. Jika Anda menggunakannya di satu bidang, Anda harus menggunakan ? di bidang lain.
- Wildcard / (garis miring) menentukan kenaikan. Di bidang menit, Anda bisa memasukkan 1/10 untuk menentukan setiap menit kesepuluh, mulai dari menit pertama jam (sebagai contoh, menit ke-11, 21, dan 31, dan seterusnya).
- Wildcard ? (tanda tanya) menentukan pilihan apa pun. Di ay-of-month bidang D Anda bisa memasukkan 7 dan jika ada hari dalam seminggu yang dapat diterima, Anda bisa masuk? di ay-of-week bidang D.
- Wildcard L di ay-of-week bidang D ay-of-month atau D menentukan hari terakhir bulan atau minggu.
- **W**Wildcard di ay-of-month bidang D menentukan hari kerja. Di ay-of-month bidang D, **3W** tentukan hari kerja yang paling dekat dengan hari ketiga bulan itu.
- Wildcard # di ay-of-week bidang D menentukan contoh tertentu dari hari yang ditentukan dalam seminggu dalam sebulan. Sebagai contoh, 3#2 akan menjadi hari Selasa kedua setiap bulan: 3 mengacu pada hari Selasa karena itu adalah hari ketiga setiap minggu, dan 2 mengacu pada hari kedua dari jenis tersebut dalam bulan tersebut.

> **Note**
>
> Jika Anda menggunakan karakter '#', Anda hanya dapat menentukan satu ekspresi di day-of-week bidang. Sebagai contoh, `"3#1,6#3"` tidak valid karena ditafsirkan sebagai dua ekspresi.

Contoh

Contoh berikut menunjukkan bagaimana menggunakan ekspresi cron dengan AWS CLI `create-schedule` perintah untuk mengkonfigurasi jadwal berbasis cron. Contoh ini membuat jadwal yang berjalan pada pukul 10:15 UTC +0 pada hari Jumat terakhir setiap bulan selama tahun 2022

hingga 2023, dan mengirimkan pesan ke SQS antrian Amazon, menggunakan jenis target templat. `SqsParameters`

```
$ aws scheduler create-schedule --schedule-expression "cron(15 10 ? * 6L 2022-2023)" --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Jadwal satu kali

Jadwal satu kali akan memanggil target hanya sekali pada tanggal dan waktu yang Anda tentukan menggunakan tanggal yang valid, dan stempel waktu. EventBridge Scheduler mendukung penjadwalan di Universal Coordinated Time (UTC), atau di zona waktu yang Anda tentukan saat Anda membuat jadwal.

> **Note**
>
> Jadwal satu kali masih dihitung terhadap kuota akun Anda setelah selesai berjalan dan menjalankan targetnya. Sebaiknya hapus jadwal satu kali Anda setelah selesai berjalan.

Anda mengonfigurasi jadwal satu kali menggunakan ekspresi at. Ekspresi at terdiri dari tanggal dan waktu di mana Anda ingin EventBridge Scheduler untuk memanggil jadwal Anda, seperti yang ditunjukkan dalam berikut ini.

Sintaks

```
at(yyyy-mm-ddThh:mm:ss)
```

Ketika Anda mengkonfigurasi jadwal satu kali, EventBridge Scheduler mengabaikan `StartDate` dan `EndDate` Anda menentukan jadwal.

Contoh

Contoh berikut menunjukkan bagaimana menggunakan pada ekspresi dengan AWS CLI `create-schedule` perintah untuk mengkonfigurasi jadwal satu kali. Contoh ini membuat jadwal yang berjalan sekali pada pukul 13.00 UTC -8 pada 20 November 2022, dan mengirimkan pesan ke SQS antrian Amazon, menggunakan tipe target template. `SqsParameters`

```
$ aws scheduler create-schedule --schedule-expression "at(2022-11-20T13:00:00)" --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--schedule-expression-timezone "America/Los_Angeles"
--flexible-time-window '{ "Mode": "OFF"}'
```

Zona waktu di EventBridge Scheduler

EventBridge Scheduler mendukung konfigurasi jadwal berbasis cron dan satu kali di zona waktu apa pun yang Anda tentukan. EventBridge Scheduler menggunakan Database Zona Waktu yang dikelola oleh Internet Assigned Numbers Authority (IANA).

Dengan AWS CLI, Anda dapat mengatur zona waktu di mana Anda ingin EventBridge Scheduler mengevaluasi jadwal Anda menggunakan `--schedule-expression-timezone` parameter. Misalnya, perintah berikut membuat jadwal berbasis cron yang memanggil SQS `SendMessage` target Amazon template di Amerika/New_York setiap hari pada pukul 8:30 pagi.

```
$ aws scheduler create-schedule --schedule-expression "cron(30 8 * * ? *)" --name
 schedule-in-est \
    --target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "This schedule runs
 in the America/New_York time zone." }' \
    --schedule-expression-timezone "America/New_York"
    --flexible-time-window '{ "Mode": "OFF"}'
```

Waktu penghematan siang hari di Scheduler EventBridge

EventBridge Scheduler secara otomatis menyesuaikan jadwal Anda untuk waktu musim panas. Ketika waktu bergeser ke depan di Musim Semi, jika ekspresi cron jatuh pada tanggal dan waktu yang tidak ada, pemanggilan jadwal Anda dilewati. Ketika waktu bergeser mundur di Musim Gugur, jadwal Anda berjalan hanya sekali dan tidak mengulangi pemanggilannya. Pemanggilan berikut terjadi secara normal pada tanggal dan waktu yang ditentukan.

EventBridge Scheduler menyesuaikan jadwal Anda tergantung pada zona waktu yang Anda tentukan saat Anda membuat jadwal. Jika Anda mengonfigurasi jadwal di America/New_York, jadwal Anda menyesuaikan kapan waktu berubah di zona waktu tersebut, sementara jadwal di Amerika/Los_Angeles disesuaikan tiga jam kemudian ketika waktu berubah di pantai barat.

Untuk jadwal berbasis tarif yang digunakan `days` sebagai unit, sepertI`rate(1 days)`, `days` mewakili durasi 24 jam pada jam. Ini berarti bahwa ketika waktu musim panas menyebabkan satu

hari memendek menjadi 23 jam, atau diperpanjang hingga 25 jam, EventBridge Scheduler masih mengevaluasi ekspresi laju 24 jam setelah pemanggilan terakhir jadwal.

> **Note**
>
> Beberapa zona waktu tidak mengamati waktu musim panas, sesuai dengan aturan dan peraturan setempat. Jika Anda membuat jadwal di zona waktu yang tidak mengamati waktu musim panas, EventBridge Scheduler tidak menyesuaikan jadwal Anda. Penyesuaian waktu siang hari tidak berlaku untuk jadwal dalam waktu terkoordinasi universal (). UTC

Contoh

Pertimbangkan skenario di mana Anda membuat jadwal menggunakan ekspresi cron berikut di America/Los_Angeles:. `cron(30 2 * * ? *)` Jadwal ini berjalan setiap hari pada pukul 2:30 pagi di zona waktu yang ditentukan.

- Musim semi ke depan — Ketika waktu bergeser ke depan di Musim Semi dari pukul 1:59 pagi hingga 3:00 pagi, EventBridge Scheduler melewatkan pemanggilan jadwal pada hari itu, dan melanjutkan menjalankan jadwal secara normal pada hari berikutnya.
- Fall-back — Ketika waktu bergeser mundur di Musim Gugur dari 2:59 pagi hingga 2:00 pagi, EventBridge Scheduler menjalankan jadwal hanya sekali pada pukul 2:30 pagi sebelum shift terjadi, tetapi tidak mengulangi pemanggilan jadwal lagi pada pukul 2:30 pagi setelah pergeseran waktu.

Mengelola jadwal di EventBridge Scheduler

Jadwal adalah sumber daya utama yang Anda buat, konfigurasikan, dan kelola menggunakan Amazon EventBridge Scheduler.

Setiap jadwal memiliki ekspresi jadwal yang menentukan kapan, dan dengan frekuensi apa, jadwal berjalan. EventBridge Scheduler mendukung tiga jenis jadwal: tarif, cron, dan jadwal satu kali. Untuk informasi selengkapnya tentang berbagai jenis jadwal, lihatJenis jadwal.

Saat Anda membuat jadwal, Anda mengonfigurasi target untuk jadwal yang akan dipanggil. Target adalah API operasi yang dipanggil EventBridge Scheduler atas nama Anda setiap kali jadwal Anda berjalan. EventBridge Scheduler mendukung dua jenis target: target template memanggil API operasi umum di seluruh grup inti layanan, dan parameter target universal (UTP) yang dapat Anda gunakan untuk memanggil lebih dari 6.000 operasi di lebih dari 270 layanan. Untuk informasi selengkapnya tentang mengonfigurasi target, lihatMengelola target.

Anda mengonfigurasi cara jadwal Anda menangani kegagalan, saat EventBridge Scheduler tidak dapat mengirimkan peristiwa dengan sukses ke target, dengan menggunakan dua mekanisme utama: kebijakan coba lagi, dan antrean huruf mati (). DLQ Kebijakan coba lagi menentukan berapa kali EventBridge Scheduler harus mencoba ulang peristiwa yang gagal, dan berapa lama untuk menyimpan peristiwa yang belum diproses. A DLQ adalah standar yang digunakan EventBridge Penjadwal SQS antrian Amazon untuk mengirimkan peristiwa yang gagal, setelah kebijakan coba lagi habis. Anda dapat menggunakan a DLQ untuk memecahkan masalah dengan jadwal Anda atau target hilirnya. Untuk informasi lebih lanjut tentang, lihatthe section called “Mengkonfigurasi DLQ”.

Di bagian ini, Anda dapat menemukan contoh untuk mengelola jadwal EventBridge Scheduler Anda menggunakan konsol, AWS CLI dan Scheduler. EventBridge SDKs

Topik

Mengubah status jadwal di EventBridge Scheduler

Jadwal EventBridge Scheduler memiliki dua status: diaktifkan dan dinonaktifkan. Contoh berikut digunakan `UpdateSchedule` untuk menonaktifkan jadwal yang menyala setiap lima menit dan memanggil target Lambda.

Saat Anda menggunakan`UpdateSchedule`, Anda harus memberikan semua parameter yang diperlukan. EventBridge Scheduler menggantikan jadwal Anda dengan informasi yang Anda berikan. Jika Anda tidak menentukan parameter yang sebelumnya Anda tetapkan, maka defaultnya. `null`

Example AWS CLI

```
$ aws scheduler update-schedule --name lambda-universal --schedule-expression 'rate(5
 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"arn:aws:scheduler:::aws-sdk:lambda:invoke"
 "Input": "{\"FunctionName\":\"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\",\"InvocationType\":\"Event\",\"Payload\":\"{\\\"message\\\":\\\"testing function\\
\"}\"}" }' \
--flexible-time-window '{ "Mode": "OFF"}' \
--state DISABLED
```

```
{
    "ScheduleArn": "arn:aws:scheduler:us-west-2:123456789012:schedule/default/lambda-
universal"
}
```

Contoh berikut menggunakan Python SDK dan `UpdateSchedule` operasi untuk menonaktifkan jadwal yang menargetkan Amazon SQS menggunakan target template.

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "{}"}

flex_window = { "Mode": "OFF" }
```

```
scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window,
    State='DISABLED')
```

Mengkonfigurasi jendela waktu fleksibel di Scheduler EventBridge

Ketika Anda mengkonfigurasi jadwal Anda dengan jendela waktu yang fleksibel, EventBridge Scheduler memanggil target dalam jendela waktu yang Anda tetapkan. Ini berguna dalam kasus yang tidak memerlukan pemanggilan target terjadwal yang tepat. Menetapkan jendela waktu yang fleksibel meningkatkan keandalan jadwal Anda dengan menyebarkan pemanggilan target Anda.

Misalnya, jika Anda mengonfigurasi jendela waktu fleksibel 15 menit untuk jadwal yang berjalan setiap jam, itu memanggil target dalam waktu 15 menit setelah waktu yang dijadwalkan. Berikut ini AWS CLI, dan SDK contoh EventBridge Scheduler digunakan `UpdateSchedule` untuk mengatur jendela waktu fleksibel 15 menit untuk jadwal yang berjalan sekali setiap jam.

> **Note**
>
> Anda harus menentukan apakah Anda ingin mengatur jendela waktu yang fleksibel atau tidak. Jika Anda tidak ingin mengatur opsi ini, tentukan`OFF`. Jika Anda mengatur nilainya`FLEXIBLE`, Anda harus menentukan jendela waktu maksimum selama jadwal Anda akan berjalan.

Example AWS CLI

```
$ aws scheduler update-schedule --name lambda-universal --schedule-expression 'rate(1
 hour)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"arn:aws:scheduler:::aws-sdk:lambda:invoke"
 "Input": "{\"FunctionName\":\"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\",\"InvocationType\":\"Event\",\"Payload\":\"{\\\"message\\\":\\\"testing function\\
\"}\"}" }' \
--flexible-time-window '{ "Mode": "FLEXIBLE", "MaximumWindowInMinutes": 15} \
```

```
{
    "ScheduleArn": "arn:aws:scheduler:us-west-2:123456789012:schedule/lambda-universal"
}
```

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "{}"}

flex_window = { "Mode": "FLEXIBLE", "MaximumWindowInMinutes": 15}

scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(1 hour)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

Mengkonfigurasi antrian surat mati jadwal di Scheduler EventBridge

Amazon EventBridge Scheduler mendukung antrian huruf mati () DLQ menggunakan Amazon Simple Queue Service. Jika jadwal gagal menjalankan targetnya, EventBridge Scheduler mengirimkan JSON muatan yang berisi detail pemanggilan dan respons apa pun yang diterima dari target ke antrian standar Amazon yang Anda tentukan. SQS

Topik berikut mengacu pada ini JSON sebagai peristiwa surat mati. Acara surat mati memungkinkan Anda memecahkan masalah dengan jadwal atau target Anda. Jika Anda mengonfigurasi kebijakan coba lagi untuk jadwal Anda, EventBridge Scheduler akan mengirimkan peristiwa surat mati yang telah menghabiskan jumlah maksimum percobaan ulang yang Anda tetapkan.

Topik berikut menjelaskan bagaimana Anda dapat mengonfigurasi SQS antrian Amazon sebagai DLQ jadwal Anda, mengatur izin yang dibutuhkan EventBridge Penjadwal untuk mengirimkan pesan ke AmazonSQS, dan menerima peristiwa surat mati dari. DLQ

Topik

Buat SQS antrian Amazon

Sebelum Anda mengkonfigurasi DLQ untuk jadwal Anda, Anda harus membuat SQS antrian Amazon standar. Untuk petunjuk cara membuat antrean menggunakan SQS konsol Amazon, lihat Membuat SQS antrian Amazon di Panduan Pengembang Layanan Antrian Sederhana Amazon.

> **Note**
>
> EventBridge Scheduler tidak mendukung penggunaan FIFO antrian sebagai jadwal Anda. DLQ

Gunakan AWS CLI perintah berikut untuk membuat antrian standar.

```
$ aws sqs create-queue --queue-name queue-name
```

Jika berhasil, Anda akan melihat `QueueURL` di output.

```
{
    "QueueUrl": "https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-dlq-test"
}
```

Setelah Anda membuat antrian, perhatikan ARN antrean. Anda akan memerlukan ARN ketika Anda menentukan jadwal EventBridge Scheduler Anda. DLQ Anda dapat menemukan antrian Anda ARN di SQS konsol Amazon, atau dengan menggunakan `get-queue-attributes` AWS CLI perintah.

```
$ aws sqs get-queue-attributes --queue-url your-dlq-url --attribute-names QueueArn
```

Jika berhasil, Anda akan melihat antrian ARN di output.

```
{
    "Attributes": {
        "QueueArn": "arn:aws:sqs:us-west-2:123456789012:scheduler-dlq-test"
    }
}
```

Di bagian berikutnya, Anda akan menambahkan izin yang diperlukan ke peran eksekusi jadwal Anda untuk memungkinkan EventBridge Scheduler mengirimkan peristiwa surat mati ke Amazon. SQS

Siapkan izin peran eksekusi

Agar EventBridge Scheduler dapat mengirimkan peristiwa surat mati ke AmazonSQS, peran eksekusi jadwal Anda memerlukan kebijakan izin berikut. Untuk informasi selengkapnya tentang melampirkan kebijakan izin baru ke peran eksekusi jadwal Anda, lihat Menyiapkan peran eksekusi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:SendMessage"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Peran eksekusi jadwal Anda mungkin sudah memiliki izin yang diperlukan jika Anda menggunakan EventBridge Scheduler untuk memanggil target Amazon. SQS API

Di bagian berikutnya, Anda akan menggunakan konsol EventBridge Scheduler dan menentukan a DLQ untuk jadwal Anda.

Tentukan antrian huruf mati

Untuk menentukanDLQ, gunakan konsol EventBridge Scheduler atau AWS CLI untuk memperbarui jadwal yang ada, atau buat yang baru.

Console

Untuk menentukan DLQ menggunakan konsol

1. Masuk ke AWS Management Console, lalu pilih tautan berikut untuk membuka bagian EventBridge Scheduler pada EventBridge conosle: home https://console.aws.amazon.com/scheduler/

2. Di konsol EventBridge Scheduler, buat jadwal baru, atau pilih jadwal yang ada dari daftar jadwal yang akan diedit.
3. Pada halaman Pengaturan, untuk antrian Dead-letter (DLQ), lakukan salah satu hal berikut:
   - Pilih Pilih SQS antrian Amazon di AWS akun saya sebagai DLQ, lalu pilih antrian ARN untuk Anda DLQ dari daftar tarik-turun.
   - Pilih Tentukan SQS antrian Amazon di AWS akun lain sebagai a DLQ, lalu masukkan antrean ARN untuk Anda. DLQ Jika Anda memilih antrian di AWS akun lain, konsol EventBridge Scheduler tidak akan dapat menampilkan antrian ARNs dalam daftar tarik-turun.
4. Tinjau pilihan Anda, lalu pilih Buat jadwal atau Simpan jadwal untuk menyelesaikan konfigurasi. DLQ
5. (Opsional) Untuk melihat DLQ detail jadwal, pilih nama jadwal dari daftar, lalu pilih tab Antrian huruf mati di halaman Detail jadwal.

AWS CLI

Untuk memperbarui jadwal yang ada menggunakan AWS CLI

- Gunakan `update-schedule`perintah untuk memperbarui jadwal Anda. Tentukan SQS antrian Amazon yang Anda buat sebelumnya sebagai. DLQ Tentukan IAM peran ARN yang Anda lampirkan SQS izin Amazon yang diperlukan sebagai peran eksekusi. Ganti semua nilai placeholder lainnya dengan informasi Anda.

```
$ aws scheduler update-schedule --name existing-schedule \
    --schedule-expression 'rate(5 minutes)' \
    --target '{"DeadLetterConfig": {"Arn": "DLQ_ARN"}, "RoleArn": "ROLE_ARN",
 "Arn":"QUEUE_ARN", "Input": "Hello world!" }' \
    --flexible-time-window '{ "Mode": "OFF"}'
```

Untuk membuat jadwal baru dengan DLQ menggunakan AWS CLI

- Gunakan `create-schedule`perintah untuk membuat jadwal. Ganti semua nilai placeholder dengan informasi Anda.

```
$ aws scheduler create-schedule --name new-schedule \
    --schedule-expression 'rate(5 minutes)' \
```

```
    --target '{"DeadLetterConfig": {"Arn": "DLQ_ARN"}, "RoleArn": "ROLE_ARN",
 "Arn":"QUEUE_ARN", "Input": "Hello world!" }' \
    --flexible-time-window '{ "Mode": "OFF"}'
```

Di bagian berikutnya, Anda akan menggunakan AWS CLI untuk menerima acara surat mati dari. DLQ

Ambil acara surat mati

Gunakan `receive-message`perintah, seperti yang ditunjukkan dalam berikut ini, untuk mengambil peristiwa huruf mati dari. DLQ Anda dapat mengatur jumlah pesan yang akan diambil menggunakan `--max-number-of-messages` atribut.

```
$ aws sqs receive-message --queue-url your-dlq-url --attribute-names All --message-
attribute-names All --max-number-of-messages 1
```

Jika berhasil, Anda akan melihat output yang mirip dengan berikut ini.

```
{
    "Messages": [
        {
            "MessageId": "2aeg3510-fe3a-4f5a-ab6a-6906560eaf7e",
            "ReceiptHandle": "AQEBkNKTdOMrWgHKPoITRBwrPoK3eCSZIcZwVqCY0BZ
+FfTcORFpopJbtCqj36VbBTlHreM8+qM/m5jcwqSlAlGmIJO/hYmMgn/
+dwIty9izE7HnpvRhhEyHxbeTZ5V05RbeasYaBdNyi9WLcnAHviDh6MebLXXNWoFyYNsxdwJuG0f/
w3htX6r3dxpXvvFNPGoQb8ihY37+u0gtsbuIwhLtUSmE8rbldEEwiUfi3IJ1zEZpUS77n/k1GWrMrnYg0Gx/
BuaLzOrFi2F738XI/
Hnh45uv3ca6OYwS1ojPQ1LtX2URg1haV5884FYlaRvY8jRlpCZabTkYRTZKSXG5KNgYZnHpmsspii6JNkjitYVFKPo0H91w
            "MD5OfBody": "07adc3fc889d6107d8bb8fda42fe0573",
            "Body": "{\"MessageBody\":\"Hello, world!",\"QueueUrl\":\"https://sqs.us-
west-2.amazonaws.com/123456789012/does-not-exist\"}",
            "Attributes": {
                "SenderId": "AROA2DZE3W4CTL5ZR7EIN:ff00212d8c453aaaae644bc6846d4723",
                "ApproximateFirstReceiveTimestamp": "1652499058144",
                "ApproximateReceiveCount": "2",
                "SentTimestamp": "1652490733042"
            },
            "MD5OfMessageAttributes": "f72c1d78100860e00403d849831d4895",
            "MessageAttributes": {
                "ERROR_CODE": {
                    "StringValue": "AWS.SimpleQueueService.NonExistentQueue",
                    "DataType": "String"
```

```
                },
                "ERROR_MESSAGE": {
                    "StringValue": "The specified queue does not exist for this wsdl
 version.",
                    "DataType": "String"
                },
                "EXECUTION_ID": {
                    "StringValue": "ad06616e51cdf74a",
                    "DataType": "String"
                },
                "EXHAUSTED_RETRY_CONDITION": {
                    "StringValue": "MaximumEventAgeInSeconds",
                    "DataType": "String"
                }
                "IS_PAYLOAD_TRUNCATED": {
                    "StringValue": "false",
                    "DataType": "String"
                },
                "RETRY_ATTEMPTS": {
                    "StringValue": "0",
                    "DataType": "String"
                },
                "SCHEDULED_TIME": {
                    "StringValue": "2022-05-14T01:12:00Z",
                    "DataType": "String"
                },
                "SCHEDULE_ARN": {
                    "StringValue": "arn:aws:scheduler:us-west-2:123456789012:schedule/
DLQ-test",
                    "DataType": "String"
                },
                "TARGET_ARN": {
                    "StringValue": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage",
                    "DataType": "String"
                }
            }
        }
    ]
}
```

Perhatikan atribut berikut dalam peristiwa dead-letter untuk membantu Anda mengidentifikasi dan memecahkan masalah kemungkinan alasan mengapa inovasi target gagal.

- **`ERROR_CODE`**— Berisi kode kesalahan yang EventBridge Scheduler terima dari layanan API target. Pada contoh sebelumnya, kode kesalahan yang dikembalikan oleh Amazon SQS adalah. `AWS.SimpleQueueService.NonExistentQueue` Jika jadwal gagal memanggil target karena masalah dengan EventBridge Scheduler, Anda akan melihat kode kesalahan berikut sebagai gantinya:. `AWS.Scheduler.InternalServerError`
- **`ERROR_MESSAGE`**— Berisi pesan kesalahan yang diterima EventBridge Scheduler dari layanan API target. Dalam contoh sebelumnya, pesan kesalahan yang dikembalikan oleh Amazon SQS adalah. `The specified queue does not exist for this wsdl version` Jika jadwal gagal karena masalah dengan EventBridge Scheduler, Anda akan melihat pesan galat berikut: `Unexpected error occurred while processing the request`
- **`TARGET_ARN`**— Target yang dipanggil jadwal Anda, dalam ARN format layanan berikut:`arn:aws:scheduler:::aws-sdk:`*`service`*`:`*`apiAction`*. ARN
- **`EXHAUSTED_RETRY_CONDITION`**— Menunjukkan mengapa acara dikirim keDLQ. Atribut ini akan hadir jika kesalahan dari target API adalah kesalahan yang dapat dicoba ulang, dan bukan kesalahan permanen. Atribut dapat berisi nilai `MaximumRetryAttempts` jika EventBridge Scheduler mengirimkannya ke DLQ setelah melebihi upaya percobaan ulang maksimum yang Anda konfigurasikan untuk jadwal, atau`MaximumEventAgeInSeconds`, jika acara lebih tua dari usia maksimum yang Anda konfigurasikan pada jadwal dan masih gagal dikirimkan.

Pada contoh sebelumnya, kita dapat menentukan, berdasarkan kode kesalahan, dan pesan kesalahan, bahwa antrian target yang kita tentukan untuk jadwal tidak ada.

Menghapus jadwal di Scheduler EventBridge

Anda dapat menghapus jadwal dengan mengonfigurasi penghapusan otomatis, atau dengan menghapus jadwal individual secara manual. Gunakan topik berikut untuk mempelajari cara menghapus jadwal menggunakan kedua metode, dan mengapa Anda dapat memilih satu metode di atas yang lain.

Topik

Penghapusan setelah jadwal selesai

Konfigurasikan penghapusan otomatis setelah jadwal selesai jika Anda ingin menghindari keharusan mengelola sumber daya jadwal Anda secara individual di EventBridge Scheduler. Dalam aplikasi di mana Anda membuat ribuan jadwal sekaligus dan membutuhkan fleksibilitas untuk meningkatkan jumlah jadwal sesuai permintaan, penghapusan otomatis dapat memastikan bahwa Anda tidak mencapai kuota akun Anda untuk jumlah jadwal di Wilayah tertentu.

Saat Anda mengonfigurasi penghapusan otomatis untuk jadwal, EventBridge Scheduler menghapus jadwal setelah pemanggilan target terakhirnya. Untuk jadwal satu kali, ini terjadi setelah jadwal telah memanggil targetnya sekali. Untuk jadwal berulang yang Anda atur dengan ekspresi rate, atau cron, jadwal Anda dihapus setelah pemanggilan terakhirnya. Pemanggilan terakhir jadwal berulang adalah pemanggilan yang terjadi paling dekat dengan yang Anda tentukan. `EndDate` Jika Anda mengkonfigurasi jadwal dengan penghapusan otomatis tetapi tidak menentukan nilai untuk`EndDate`, EventBridge Scheduler tidak secara otomatis menghapus jadwal.

Anda dapat mengatur penghapusan otomatis saat pertama kali membuat jadwal, atau memperbarui preferensi untuk jadwal yang ada. Langkah-langkah berikut menjelaskan cara mengonfigurasi penghapusan otomatis untuk jadwal yang ada.

AWS Management Console

1. Buka konsol EventBridge Scheduler di https://console.aws.amazon.com/scheduler/.
2. Dari daftar jadwal, pilih jadwal yang ingin Anda edit, lalu pilih Edit.
3. Dari daftar navigasi di sebelah kiri, pilih Pengaturan.
4. Di bagian Tindakan setelah jadwal selesai, pilih DELETEdari daftar drop-down, lalu simpan perubahan Anda.

AWS CLI

1. Buka jendela prompt baru.
2. Gunakan AWS CLI perintah update-schedule untuk memperbarui jadwal yang ada yang ditunjukkan di berikut ini. Perintah menetapkan `--action-after-completion` ke`DELETE`. Contoh ini mengasumsikan bahwa Anda telah menentukan konfigurasi target Anda secara lokal dalam sebuah JSON file. Untuk memperbarui jadwal, Anda harus memberikan target, serta parameter jadwal lainnya yang ingin Anda konfigurasikan untuk jadwal yang ada.

Ini adalah jadwal berulang dengan tingkat satu doa per jam. Oleh karena itu, Anda menentukan tanggal akhir saat mengatur `--action-after-completion` parameter.

```
$ aws scheduler update-schedule --name schedule-name
 \
--action-after-completion 'DELETE' \
--schedule-expression 'rate(1 hour)' \
--end-date '2024-01-01T00:00:00'
--target file://target-configuration.json \
--flexible-time-window '{ "Mode": "OFF"}' \
```

Penghapusan manual

Ketika Anda tidak lagi membutuhkan jadwal, Anda dapat menghapusnya menggunakan `DeleteSchedule`operasi.

Example AWS CLI

```
$ aws scheduler delete-schedule --name your-schedule
```

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

scheduler.delete_schedule(Name="your-schedule")
```

Apa selanjutnya?

- Untuk informasi selengkapnya tentang cara mengonfigurasi target template untuk Lambda dan Step Functions, dan untuk mempelajari cara menggunakan parameter target universal, lihat. Mengelola target
- Untuk informasi selengkapnya tentang tipe dan API operasi data EventBridge Scheduler, lihat Referensi EventBridge Penjadwal API.

Mengelola grup jadwal di EventBridge Scheduler

Grup jadwal adalah sumber daya Amazon EventBridge Scheduler yang Anda gunakan untuk mengatur jadwal Anda.

Anda Akun AWS datang dengan grup `default` penjadwal. Anda dapat mengaitkan jadwal baru dengan `default` grup atau dengan grup jadwal yang Anda buat dan kelola. Anda dapat membuat hingga 500 grup jadwal di Akun AWS. Dengan EventBridge Scheduler, Anda mengatur grup jadwal, bukan jadwal individual, dengan menerapkan tag.

Tag adalah label yang terdiri dari kunci case-sensitive dan nilai case-sensitive yang Anda tentukan. Anda dapat membuat tag untuk mengkategorikan jadwal berdasarkan kriteria seperti tujuan, pemilik, atau lingkungan. Misalnya, Anda dapat mengidentifikasi lingkungan tempat jadwal Anda berada dengan tag berikut:`environment:production`.

> **Important**
>
> Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam tag. Tag dapat diakses oleh banyak AWS layanan, termasuk penagihan. Tag tidak dimaksudkan untuk digunakan dalam data sensitif atau privat.

Grup jadwal memiliki dua kemungkinan keadaan: ACTIVEdan DELETING.

Ketika Anda pertama kali membuat grup, itu secara `ACTIVE` default. Anda dapat menambahkan jadwal ke `ACTIVE` grup. Saat Anda menghapus grup, status berubah `DELETING` hingga EventBridge Scheduler menyelesaikan penghapusan jadwal terkait. Setelah EventBridge Scheduler menghapus jadwal dalam grup, grup tidak lagi tersedia di akun Anda.

Gunakan topik berikut untuk membuat grup jadwal dan menerapkan tag untuk itu. Anda juga akan mengaitkan jadwal dengan grup. Akhirnya, Anda akan menghapus grup.

Topik

Membuat grup jadwal di EventBridge Scheduler

Gunakan grup jadwal dan penandaan untuk mengatur jadwal yang memiliki tujuan bersama atau milik lingkungan yang sama. Pada langkah-langkah berikut, Anda membuat grup jadwal baru dan memberi label menggunakan tag. Anda kemudian mengaitkan jadwal baru dengan grup itu.

> **Note**
>
> Setelah membuat grup, Anda tidak dapat menghapus jadwal dari grup tersebut, atau mengaitkan jadwal dengan grup yang berbeda. Anda hanya dapat mengaitkan jadwal dengan grup saat pertama kali membuat jadwal.

Langkah satu: Buat grup jadwal baru

Topik berikut menjelaskan cara membuat grup jadwal baru dan memberi label dengan tag berikut:`environment:development`.

AWS Management Console

Untuk membuat grup baru menggunakan AWS Management Console

1. Masuk ke AWS Management Console dan buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Di panel navigasi kiri, pilih Jadwalkan grup.
3. Pada halaman Jadwal grup, pilih Buat grup jadwal.
4. Di bagian Jadwal detail grup, untuk Nama, masukkan nama untuk grup. Misalnya, **`TestGroup`**.
5. Di bagian Tag, lakukan hal berikut:
   a. Pilih Tambahkan tag baru.
   b. Untuk Kunci, masukkan nama yang ingin Anda tetapkan ke kunci ini. Untuk tutorial ini, untuk memberi label pada lingkungan yang dimiliki grup jadwal ini, masukkan**`environment`**.
   c. Untuk Nilai - opsional, masukkan nilai yang ingin Anda tetapkan ke kunci ini. Untuk tutorial ini, masukkan nilai **`development`** untuk kunci lingkungan Anda.

> **Note**
>
> Anda dapat menambahkan tag tambahan ke grup Anda setelah Anda membuatnya.

6. Untuk menyelesaikan, pilih Buat grup jadwal. Grup baru Anda muncul di daftar Jadwal grup.
7. (Opsional) Untuk mengedit grup atau mengelola tagnya, pilih kotak centang untuk grup baru dan pilih Edit.

> **Note**
>
> Anda tidak dapat mengedit grup `default` jadwal.

AWS CLI

Untuk membuat grup baru menggunakan AWS CLI

1. Buka jendela prompt perintah baru.
2. Dari AWS Command Line Interface (AWS CLI), masukkan `create-schedule-group`perintah berikut untuk membuat grup baru. Perintah ini membuat grup dengan satu tag:`environment:development`. Anda dapat menggunakan tag ini atau sistem penandaan serupa untuk memberi label pada grup jadwal Anda sesuai dengan lingkungan tempat mereka berada.

   Ganti nama jadwal dan kunci tag dan nilai dengan informasi Anda.

   ```
   $ aws scheduler create-schedule-group --name TestGroup --tags
    Key=environment,Value=development
   ```

Secara default, grup baru Anda ada di `ACTIVE` negara bagian. Anda sekarang dapat mengaitkan jadwal baru dengan grup baru yang Anda buat.

Langkah kedua: Mengaitkan jadwal dengan grup

Gunakan langkah-langkah berikut untuk mengaitkan jadwal baru dengan grup yang Anda buat di langkah sebelumnya.

AWS Management Console

Untuk mengaitkan jadwal dengan grup menggunakan AWS Management Console

1. Masuk ke AWS Management Console dan buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Di panel navigasi kiri, pilih Jadwal di panel navigasi kiri.
3. Dari tabel Jadwal, pilih Buat jadwal untuk membuat jadwal baru.
4. Pada halaman Tentukan detail jadwal, untuk Jadwal grup, pilih nama grup baru Anda dari daftar drop-down. Misalnya, pilih`TestGroup`.
5. Tentukan pola jadwal, target, pengaturan lalu tinjau pilihan Anda di halaman Tinjau dan simpan jadwal. Untuk informasi selengkapnya tentang mengonfigurasi jadwal baru, lihatMemulai.
6. Untuk menyelesaikan dan menyimpan jadwal Anda, pilih Simpan jadwal.

AWS CLI

Untuk mengaitkan jadwal dengan grup menggunakan AWS CLI

1. Buka jendela prompt perintah baru.
2. Dari AWS Command Line Interface (AWS CLI), masukkan `create-schedule`perintah berikut. Ini membuat jadwal dan mengaitkannya dengan grup dari langkah sebelumnya, bernama`sqs-test-schedule`. Jadwal ini menggunakan jenis SQS target Amazon template untuk menjalankan operasi. `SendMessage` Ganti nama jadwal, target, dan nama grup dengan informasi Anda.

```
$ aws scheduler create-schedule --name sqs-test-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }'
 \
--group-name TestGroup
--flexible-time-window '{ "Mode": "OFF"}'
```

Jadwal baru Anda sekarang dikaitkan dengan grup `TestGroup` jadwal.

Menghapus grup jadwal di EventBridge Scheduler

Berikut ini, Anda dapat mempelajari cara menghapus grup jadwal menggunakan AWS Management Console dan AWS Command Line Interface. Saat Anda menghapus grup, grup tersebut berada dalam `DELETING` status hingga EventBridge Scheduler menghapus semua jadwal dalam grup. Setelah EventBridge Scheduler menghapus jadwal dalam grup, grup tidak lagi tersedia di akun Anda.

> **Note**
>
> Setelah membuat grup, Anda tidak dapat menghapus jadwal dari grup tersebut, atau mengaitkan jadwal dengan grup yang berbeda. Anda hanya dapat mengaitkan jadwal dengan grup saat pertama kali membuat jadwal.

AWS Management Console

Untuk menghapus grup menggunakan AWS Management Console

1. Masuk ke AWS Management Console dan buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Di panel navigasi kiri, pilih Jadwalkan grup di panel navigasi kiri.
3. Pada halaman Jadwal grup, dari daftar grup yang ada di saat ini Wilayah AWS, cari grup yang ingin Anda hapus. Jika Anda tidak melihat grup yang Anda cari, pilih yang lain Wilayah AWS.

   > **Note**
   >
   > Anda tidak dapat menghapus, atau mengedit, grup default.

4. Pilih kotak centang untuk grup yang ingin Anda hapus.
5. Pilih Hapus.
6. Dalam kotak dialog Hapus jadwal grup, masukkan nama grup untuk mengonfirmasi pilihan Anda, lalu pilih Hapus.
7. Dalam daftar Grup jadwal, kolom Status berubah untuk menunjukkan bahwa grup Anda sekarang Menghapus. Grup tetap dalam keadaan ini sampai EventBridge Scheduler menghapus semua jadwal yang terkait dengan grup.
8. Untuk menyegarkan daftar dan mengonfirmasi bahwa grup telah dihapus, pilih ikon Refresh.

AWS CLI

Untuk menghapus grup menggunakan AWS CLI

1. Buka jendela prompt perintah baru.
2. Dari AWS Command Line Interface (AWS CLI), masukkan `delete-schedule-group`perintah berikut untuk menghapus grup jadwal. Ganti nilainya `--name` dengan informasi Anda.

```
$ aws scheduler delete-schedule-group --name TestGroup
```

   Jika berhasil, AWS CLI operasi ini tidak mengembalikan respons.

3. Untuk memverifikasi bahwa grup berada dalam `DELETING` status, jalankan `get-schedule-group`perintah berikut.

```
$ aws scheduler get-schedule-group --name TestGroup
```

   Jika berhasil, Anda menerima output yang mirip dengan berikut ini:

```
{
    "Arn": "arn:aws::scheduler:us-west-2:123456789012:schedule-group/TestGroup",
    "CreationDate": "2023-01-01T09:00:00.000000-07:00",
    "LastModificationDate": "2023-01-01T09:00:00.000000-07:00",
    "Name": "TestGroup",
    "State": "DELETING"
}
```

   EventBridge Scheduler menghapus grup setelah menghapus jadwal yang terkait dengan grup. Jika Anda menjalankan `get-schedule-group` lagi, Anda menerima `ResourceNotFoundException` tanggapan berikut:

```
An error occurred (ResourceNotFoundException) when calling the GetScheduleGroup
 operation: Schedule group TestGroup does not exist.
```

Sumber daya terkait

Untuk informasi selengkapnya tentang grup jadwal, lihat sumber daya berikut:

- CreateScheduleGroupoperasi di APIReferensi EventBridge Penjadwal.
- DeleteScheduleGroupoperasi di APIReferensi EventBridge Penjadwal.

Mengelola target di EventBridge Scheduler

Topik berikut menjelaskan cara menggunakan target template dan universal dengan EventBridge Scheduler, dan menyediakan daftar AWS layanan yang didukung yang dapat Anda konfigurasi menggunakan parameter target universal EventBridge Scheduler.

Target Templated adalah serangkaian API operasi umum di sekelompok AWS layanan inti seperti AmazonSQS, Lambda, dan Step Functions. Misalnya, Anda dapat menargetkan API operasi Invoke Lambda dengan menyediakan fungsiARN, atau `SendMessage`operasi SQS Amazon dengan antrian ARN target.

Target universal adalah serangkaian parameter yang dapat disesuaikan yang memungkinkan Anda untuk memanggil serangkaian API operasi yang lebih luas untuk banyak layanan. AWS Misalnya, Anda dapat menggunakan parameter target universal EventBridge Scheduler (UTP) untuk membuat SQS antrean Amazon baru menggunakan operasi. `CreateQueue`

Untuk mengonfigurasi target template atau universal, jadwal Anda harus memiliki izin untuk memanggil API operasi yang Anda konfigurasikan sebagai target Anda. Anda melakukan ini dengan melampirkan izin yang diperlukan untuk peran eksekusi jadwal Anda. Misalnya, untuk menargetkan `SendMessage`operasi SQS Amazon, peran eksekusi diberikan izin untuk melakukan `sqs:SendMessage` tindakan. Dalam kebanyakan kasus, Anda dapat menambahkan izin yang diperlukan dengan menggunakan kebijakan AWS terkelola yang didukung oleh layanan target. Namun, Anda juga dapat membuat kebijakan terkelola pelanggan Anda sendiri, atau menambahkan izin sebaris ke kebijakan yang ada yang dilampirkan pada peran eksekusi. Topik berikut menunjukkan contoh penambahan izin untuk tipe target template dan universal.

Untuk informasi selengkapnya tentang menyiapkan peran eksekusi untuk jadwal, lihatthe section called “Mengatur peran eksekusi”.

Topik

Menggunakan target template di Scheduler EventBridge

Target Templated adalah serangkaian API operasi umum di sekelompok AWS layanan inti, seperti Amazon, LambdaSQS, dan Step Functions. Misalnya, Anda dapat menargetkan `Invoke`operasi Lambda dengan menyediakan fungsiARN, atau `SendMessage`operasi SQS Amazon menggunakan antrianARN. Untuk mengonfigurasi target template, Anda juga harus memberikan izin ke peran eksekusi jadwal untuk melakukan operasi yang ditargetkanAPI.

Untuk mengonfigurasi target template secara terprogram menggunakan AWS CLI atau salah satu dari EventBridge SchedulerSDKs, Anda perlu menentukan peran eksekusi, sumber daya target ARN untuk, input opsional yang ingin Anda kirimkan oleh EventBridge Scheduler ke target, dan untuk beberapa target template, serangkaian parameter unik dengan opsi konfigurasi tambahan untuk target tersebut. ARN Saat Anda menentukan sumber daya target template, EventBridge Scheduler secara otomatis mengasumsikan bahwa Anda ingin memanggil API operasi yang didukung untuk layanan tersebut. ARN Jika Anda ingin EventBridge Scheduler menargetkan API operasi yang berbeda untuk layanan, Anda harus mengonfigurasi target sebagai target universal.

Berikut ini adalah daftar lengkap semua target template yang didukung EventBridge Scheduler, dan jika berlaku, setiap set unik parameter terkait target. Pilih tautan untuk setiap set parameter untuk melihat bidang wajib, dan opsional, di APIReferensi EventBridge Penjadwal.

- CodeBuild – `StartBuild`
- CodePipeline – `StartPipelineExecution`
- Amazon ECS — `RunTask`
  - Parameter: `EcsParameters`
- EventBridge – `PutEvents`
  - Parameter: `EventBridgeParameters`
- Amazon Inspector - `StartAssessmentRun`
- Kinesis — `PutRecord`
  - Parameter: `KinesisParameters`
- Firehose — `PutRecord`
- Lambda – `Invoke`
- SageMaker – `StartPipelineExecution`
  - Parameter: `SageMakerPipelineParameters`
- Amazon SNS — `Publish`

- Amazon SQS — SendMessage
  - Parameter: SqsParameters
- Step Functions - StartExecution

Gunakan contoh berikut untuk mempelajari cara mengonfigurasi target template yang berbeda, dan IAM izin yang diperlukan untuk setiap target yang dijelaskan.

Amazon SQS **SendMessage**

Example Kebijakan izin untuk peran eksekusi

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:SendMessage"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Example AWS CLI

```
$ aws scheduler create-schedule --name sqs-templated --schedule-expression 'rate(5
 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"QUEUE_ARN", "Input": "Message for scheduleArn:
 '<aws.scheduler.schedule-arn>', scheduledTime: '<aws.scheduler.scheduled-time>'" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sqs_templated = {
```

```
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "Message for scheduleArn: '<aws.scheduler.schedule-arn>', scheduledTime:
 '<aws.scheduler.scheduled-time>'"
}

scheduler.create_schedule(
    Name="sqs-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

Example Jawa SDK

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target sqsTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<QUEUE_ARN>")
                .input("Message for scheduleArn: '<aws.scheduler.schedule-arn>',
 scheduledTime: '<aws.scheduler.scheduled-time>'")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(sqsTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
```

```
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and an Amazon SQS
 templated target");
    }
}
```

Lambda **Invoke**

Example Kebijakan izin untuk peran eksekusi

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Example AWS CLI

```
$ aws scheduler create-schedule --name lambda-templated-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"FUNCTION_ARN", "Input": "{ \"Payload\":
 \"TEST_PAYLOAD\" }" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

lambda_templated = {
    "RoleArn": "<ROLE_ARN>",
```

```
    "Arn": "<LAMBDA_ARN>",
    "Input": "{ 'Payload': 'TEST_PAYLOAD' }"}
}

scheduler.create_schedule(
    Name="lambda-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=lambda_templated,
    FlexibleTimeWindow=flex_window)
```

Example Jawa SDK

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target lambdaTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<Lambda ARN>")
                .input("{ 'Payload': 'TEST_PAYLOAD' }")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE_NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(lambdaTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .clientToken("<Token GUID>")
                .build();
```

```
        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Lambda templated
 target");
    }
}
```

Step Functions **`StartExecution`**

Example Kebijakan izin untuk peran eksekusi

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "states:StartExecution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Example AWS CLI

```
$ aws scheduler create-schedule --name sfn-templated-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"STATE_MACHINE_ARN", "Input": "{ \"Payload\":
 \"TEST_PAYLOAD\" }" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sfn_templated= {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<STATE_MACHINE_ARN>",
    "Input": "{ 'Payload': 'TEST_PAYLOAD' }"
```

```
}

scheduler.create_schedule(Name="sfn-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sfn_templated,
    FlexibleTimeWindow=flex_window)
```

Example Jawa SDK

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target stepFunctionsTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<STATE_MACHINE_ARN>")
                .input("{ 'Payload': 'TEST_PAYLOAD' }")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE_NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(stepFunctionsTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .clientToken("<Token GUID>")
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Step Function
 templated target");
```

```
    }
}
```

Menggunakan target universal di EventBridge Scheduler

Target universal adalah serangkaian parameter yang dapat disesuaikan yang memungkinkan Anda untuk memanggil serangkaian API operasi yang lebih luas untuk banyak layanan. AWS Misalnya, Anda dapat menggunakan parameter target universal (UTP) untuk membuat SQS antrian Amazon baru menggunakan `CreateQueue`operasi.

Untuk mengonfigurasi target universal untuk jadwal Anda menggunakan AWS CLI, atau salah satu EventBridge PenjadwalSDKs, Anda perlu menentukan informasi berikut:

- RoleArn— ARN Untuk peran eksekusi yang ingin Anda gunakan untuk target. Peran eksekusi yang Anda tentukan harus memiliki izin untuk memanggil API operasi yang Anda ingin jadwal Anda targetkan.
- Arn — Layanan lengkapARN, termasuk API operasi yang ingin Anda targetkan, dalam format berikut:`arn:aws:scheduler:::aws-sdk:`*`service`*`:`*`apiAction`*.

  Misalnya, untuk AmazonSQS, nama layanan yang Anda tentukan adalah`arn:aws:scheduler:::aws-sdk:`**`sqs:sendMessage`**.
- Input — Sebuah bentuk yang JSON Anda tentukan dengan baik dengan parameter permintaan yang EventBridge Scheduler kirim ke target. API Parameter dan bentuk yang JSON Anda tetapkan `Input` ditentukan oleh layanan yang API dipanggil jadwal Anda. Untuk menemukan informasi ini, lihat API referensi untuk layanan yang ingin Anda targetkan.

Tindakan yang tidak didukung

EventBridge Scheduler tidak mendukung API tindakan hanya-baca, seperti `GET` operasi umum, yang dimulai dengan daftar awalan berikut:

```
get
describe
list
poll
receive
search
scan
```

```
query
select
read
lookup
discover
validate
batchGet
batchDescribe
batchRead
transactGet
adminGet
adminList
testMigration
retrieve
testConnection
translateDocument
isAuthorized
invokeModel
```

Misalnya, layanan ARN untuk `GetQueueUrl`APItindakan tersebut adalah sebagai berikut:`arn:aws:scheduler:::aws-sdk:sqs:`**`getQueueURL`**. Karena API tindakan dimulai dengan `get` awalan, EventBridge Scheduler tidak mendukung target ini. Demikian pula, `ListBrokers`tindakan Amazon MQ tidak didukung sebagai target karena operasi dimulai dengan awalan. `list`

Contoh menggunakan target universal

Parameter yang Anda lewati di `Input` bidang jadwal bergantung pada parameter permintaan yang diterima oleh layanan yang ingin API Anda panggil. Misalnya, untuk menargetkan Lambda `Invoke`, Anda dapat mengatur parameter yang tercantum dalam AWS Lambda API Referensi. Ini termasuk JSON payload opsional yang dapat Anda berikan ke fungsi Lambda.

Untuk menentukan parameter yang dapat Anda atur untuk berbedaAPIs, lihat API referensi untuk layanan tersebut. Mirip dengan Lambda`Invoke`, beberapa APIs menerima URI parameter, serta payload badan permintaan. Dalam kasus seperti itu, Anda menentukan parameter URI jalur serta JSON muatan dalam jadwal `Input` Anda.

Contoh berikut menunjukkan cara Anda menggunakan target universal untuk menjalankan API operasi umum dengan Lambda, SQS Amazon, dan Step Functions.

Example Lambda

```
$ aws scheduler create-schedule --name lambda-universal-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"arn:aws:scheduler:::aws-sdk:lambda:invoke"
 "Input": "{\"FunctionName\":\"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\",\"InvocationType\":\"Event\",\"Payload\":\"{\\\"message\\\":\\\"testing function\\
\"}\"}" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example Amazon SQS

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sqs_universal= {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage",
    "Input": "{\"MessageBody\":\"My message\",\"QueueUrl\":\"<QUEUE_URL>\"}"}
}

scheduler.create_schedule(
    Name="sqs-sdk-test",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_universal,
    FlexibleTimeWindow=flex_window)
```

Example Step Functions

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {
```

```
        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target stepFunctionsUniversalTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("arn:aws:scheduler:::aws-sdk:sfn:startExecution")
                .input("{\"Input\":\"{}\",\"StateMachineArn\":\"<STATE_MACHINE_ARN>
\"}")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE_NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(stepFunctionsUniversalTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .clientToken("<Token GUID>")
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Step Function
 universal target");
    }
}
```

Menambahkan atribut konteks di EventBridge Scheduler

Penggunaan kata kunci berikut di payload yang Anda berikan ke target untuk mengumpulkan metadata tentang jadwal. EventBridge Scheduler menggantikan setiap kata kunci dengan nilainya masing-masing saat jadwal Anda memanggil target.

- **`<aws.scheduler.schedule-arn>`**- Jadwal. ARN
- **`<aws.scheduler.scheduled-time>`**— Waktu yang Anda tentukan untuk jadwal untuk memanggil targetnya, misalnya,`2022-03-22T18:59:43Z`.
- **`<aws.scheduler.execution-id>`**— ID unik yang ditetapkan oleh EventBridge Scheduler untuk setiap percobaan pemanggilan target, misalnya,. `d32c5kddcf5bb8c3`
- **`<aws.scheduler.attempt-number>`**— Penghitung yang mengidentifikasi nomor percobaan untuk pemanggilan saat ini, misalnya,. 1

Contoh ini menunjukkan pembuatan jadwal yang menyala setiap lima menit, dan memanggil SQS `SendMessage` operasi Amazon sebagai target universal. Badan pesan mencakup nilai untuk`schedule-time`.

Example AWS CLI

```
$ aws scheduler create-schedule --name your-schedule \
    --schedule-expression 'rate(5 minutes)' \
    --target '{"RoleArn": "ROLE_ARN", \
        "Arn": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage", \
        "Input": "{\"MessageBody\":\"<aws.scheduler.scheduled-time>\",\"QueueUrl\":
\"https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-cli-test\"}"}' \
    --flexible-time-window '{ "Mode": "OFF"}'
```

Example Python SDK

```
import boto3
scheduler = boto3.client('scheduler')

sqs_universal= {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage",
    "Input": "{\"MessageBody\":\"<aws.scheduler.scheduled-time>\",\"QueueUrl\":
\"https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-cli-test\"}"
}

flex_window = { "Mode": "OFF" }


scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_universal,
    FlexibleTimeWindow=flex_window)
```

Apa selanjutnya?

Untuk informasi selengkapnya tentang tipe dan API operasi data EventBridge Scheduler, lihat Referensi EventBridge Penjadwal API.

Keamanan di Amazon EventBridge Scheduler

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon EventBridge Scheduler, lihat AWS Layanan dalam Lingkup berdasarkan AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan EventBridge Scheduler. Topik berikut menunjukkan cara mengkonfigurasi EventBridge Scheduler untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya EventBridge Penjadwal Anda.

Topik

Mengelola akses ke Amazon EventBridge Scheduler

AWS Identity and Access Management (IAM) adalah AWS layanan yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. IAMadministrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan EventBridge sumber Penjadwal. IAMadalah AWS layanan yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di EventBridge Scheduler.

Pengguna layanan — Jika Anda menggunakan layanan EventBridge Scheduler untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur EventBridge Penjadwal untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di EventBridge Scheduler, lihatMemecahkan masalah identitas dan akses Amazon EventBridge Scheduler.

Administrator layanan — Jika Anda bertanggung jawab atas sumber EventBridge Scheduler di perusahaan Anda, Anda mungkin memiliki akses penuh ke EventBridge Scheduler. Tugas Anda adalah menentukan fitur dan sumber daya EventBridge Penjadwal mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasarIAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat

menggunakan IAM EventBridge Scheduler, lihatBagaimana EventBridge Scheduler bekerja dengan IAM.

IAMadministrator - Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke EventBridge Scheduler. Untuk melihat contoh kebijakan berbasis identitas EventBridge Scheduler yang dapat Anda gunakan, lihat. IAM Menggunakan kebijakan berbasis identitas di Scheduler EventBridge

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai IAM pengguna, atau dengan mengambil peranIAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (Pusat IAM Identitas), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani AWS API permintaan di Panduan IAM Pengguna.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat Autentikasi multi-faktor di Panduan AWS IAM Identity Center Pengguna dan Menggunakan autentikasi multi-faktor (MFA) AWS di Panduan Pengguna. IAM

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua AWS layanan dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensi pengguna root di IAMPanduan Pengguna.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS layanan dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses AWS layanan dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat IAM Identitas, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat IAM Identitas, lihat Apa itu Pusat IAM Identitas? dalam AWS IAM Identity Center User Guide.

Pengguna dan grup IAM

IAMPengguna adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya mengandalkan kredensi sementara daripada membuat IAM pengguna yang memiliki kredensi jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensil jangka panjang dengan IAM pengguna, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensi jangka panjang di IAMPanduan Pengguna.

IAMGrup adalah identitas yang menentukan kumpulan IAM pengguna. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup bernama IAMAdminsdan memberikan izin grup tersebut untuk mengelola sumber dayaIAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari lebih lanjut, lihat Kapan membuat IAM pengguna (bukan peran) di Panduan IAM Pengguna.

IAMperan

IAMPeran adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Ini mirip dengan IAM pengguna, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil IAM peran sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil AWS CLI atau AWS API operasi atau dengan menggunakan kustomURL. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat Menggunakan IAM peran di Panduan IAM Pengguna.

IAMperan dengan kredensi sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga di Panduan IAM Pengguna. Jika Anda menggunakan Pusat IAM Identitas, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, Pusat IAM Identitas menghubungkan izin yang disetel ke peran. IAM Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin IAM pengguna sementara — IAM Pengguna atau peran dapat mengambil IAM peran untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun — Anda dapat menggunakan IAM peran untuk memungkinkan seseorang (prinsipal tepercaya) di akun lain mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa AWS layanan, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai

proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM Panduan Pengguna. IAM

- Akses lintas layanan — Beberapa AWS layanan menggunakan fitur lain AWS layanan. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FASmenggunakan izin dari pemanggilan utama AWS layanan, dikombinasikan dengan permintaan AWS layanan untuk membuat permintaan ke layanan hilir. FASPermintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain AWS layanan atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat Meneruskan sesi akses.
  - Peran layanan — Peran layanan adalah IAMperan yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAMAdministrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalamIAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke AWS layanan dalam IAMPanduan Pengguna.
  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. AWS layanan Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredenal sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS API meminta. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di IAMPanduan Pengguna.

Untuk mempelajari apakah akan menggunakan IAM peran atau IAM pengguna, lihat Kapan membuat IAM peran (bukan pengguna) di Panduan IAM Pengguna.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai JSON dokumen. Untuk informasi selengkapnya tentang struktur dan isi dokumen JSON kebijakan, lihat Ringkasan JSON kebijakan di Panduan IAM Pengguna.

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

IAMkebijakan menentukan izin untuk tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasi. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan itu bisa mendapatkan informasi peran dari AWS Management Console, AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat IAM kebijakan di Panduan Pengguna. IAM

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan sebaris, lihat Memilih antara kebijakan terkelola dan kebijakan sebaris di IAMPanduan Pengguna.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. AWS layanan

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLsmirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnyaACLs, lihat Ikhtisar daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batas izin — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas (pengguna atau peran). IAM IAM Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batas izin, lihat Batas izin untuk IAM entitas di IAMPanduan Pengguna.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah JSON kebijakan yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations

adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCPMembatasi izin untuk entitas di akun anggota, termasuk masing-masing Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations danSCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan secara tegas dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat Kebijakan sesi di Panduan IAM Pengguna.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan IAM Pengguna.

Bagaimana EventBridge Scheduler bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke EventBridge Scheduler, pelajari IAM fitur apa saja yang tersedia untuk digunakan dengan EventBridge Scheduler.

IAMfitur yang dapat Anda gunakan dengan Amazon EventBridge Scheduler

IAMfitur	EventBridge Dukungan penjadwal
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya

IAMfitur	EventBridge Dukungan penjadwal
ACLs	Tidak
ABAC(tag dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin prinsipal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja EventBridge Penjadwal dan AWS layanan lainnya dengan sebagian besar IAM fitur, lihat AWS layanan yang berfungsi IAM di IAMPanduan Pengguna.

Kebijakan berbasis identitas untuk Scheduler EventBridge

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat IAM kebijakan di Panduan Pengguna. IAM

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam JSON kebijakan, lihat referensi elemen IAM JSON kebijakan di Panduan IAM Pengguna.

Contoh kebijakan berbasis identitas untuk Scheduler EventBridge

Untuk melihat contoh kebijakan berbasis identitas EventBridge Scheduler, lihat. Menggunakan kebijakan berbasis identitas di Scheduler EventBridge

Kebijakan berbasis sumber daya dalam Scheduler EventBridge

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. AWS layanan

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau IAM entitas di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, IAM administrator di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun IAM di Panduan IAM Pengguna.

Tindakan kebijakan untuk EventBridge Scheduler

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

`Action`Elemen JSON kebijakan menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan AWS API operasi terkait. Ada beberapa pengecualian, seperti tindakan khusus izin yang tidak memiliki operasi yang cocok. API Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan EventBridge Penjadwal, lihat Tindakan yang ditentukan oleh EventBridge Penjadwal Amazon di Referensi Otorisasi Layanan.

Tindakan kebijakan di EventBridge Scheduler menggunakan awalan berikut sebelum tindakan:

```
scheduler
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [
      "scheduler:action1",
      "scheduler:action2"
]
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `List`, sertakan tindakan berikut:

```
"Action": [
   "scheduler:List*"
]
```

Sumber daya kebijakan untuk EventBridge Scheduler

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Resource` JSON kebijakan menentukan objek atau objek yang tindakan tersebut berlaku. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Sebagai praktik terbaik, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya EventBridge Penjadwal dan jenisnyaARNs, lihat Sumber daya yang ditentukan oleh Amazon EventBridge Scheduler di Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang ditentukan oleh Amazon EventBridge Scheduler.

Untuk melihat contoh kebijakan berbasis identitas EventBridge Scheduler, lihat. Menggunakan kebijakan berbasis identitas di Scheduler EventBridge

Kunci kondisi kebijakan untuk EventBridge Scheduler

Mendukung kunci kondisi kebijakan khusus layanan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin IAM pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama IAM pengguna mereka. Untuk informasi selengkapnya, lihat elemen IAM kebijakan: variabel dan tag di Panduan IAM Pengguna.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan IAM Pengguna.

Untuk melihat daftar kunci kondisi EventBridge Penjadwal, lihat Kunci kondisi untuk EventBridge Penjadwal Amazon di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh Amazon EventBridge Scheduler.

Untuk melihat contoh kebijakan berbasis identitas EventBridge Scheduler, lihat. Menggunakan kebijakan berbasis identitas di Scheduler EventBridge

ACLsdi EventBridge Scheduler

MendukungACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLsmirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

ABACdengan EventBridge Scheduler

Mendukung ABAC (tag dalam kebijakan): Sebagian

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke IAM entitas (pengguna atau peran) dan ke banyak AWS sumber daya. Menandai entitas dan sumber daya adalah langkah pertama dari. ABAC Kemudian Anda merancang ABAC kebijakan untuk mengizinkan operasi ketika tag prinsipal cocok dengan tag pada sumber daya yang mereka coba akses.

ABACmembantu dalam lingkungan yang berkembang pesat dan membantu dengan situasi di mana manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi lebih lanjut tentangABAC, lihat Apa ituABAC? dalam IAMUser Guide. Untuk melihat tutorial dengan langkah-langkah penyiapanABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) di IAMPanduan Pengguna.

Menggunakan kredensil sementara dengan Scheduler EventBridge

Mendukung kredensi sementara: Ya

Beberapa AWS layanan tidak berfungsi saat Anda masuk menggunakan kredensil sementara. Untuk informasi tambahan, termasuk yang AWS layanan bekerja dengan kredensil sementara, lihat AWS layanan yang berfungsi IAM di IAMPanduan Pengguna.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan link sign-on (SSO) tunggal perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang beralih peran, lihat Beralih ke peran (konsol) di Panduan IAM Pengguna.

Anda dapat secara manual membuat kredensil sementara menggunakan atau. AWS CLI AWS API Anda kemudian dapat menggunakan kredensil sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensi keamanan sementara di. IAM

Izin utama lintas layanan untuk Scheduler EventBridge

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FASmenggunakan izin dari pemanggilan utama AWS layanan, dikombinasikan dengan permintaan AWS layanan untuk membuat permintaan ke layanan hilir. FASPermintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain AWS layanan atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat Meneruskan sesi akses.

Peran layanan untuk EventBridge Scheduler

Mendukung peran layanan: Ya

Peran layanan adalah IAMperan yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAMAdministrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalamIAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke AWS layanan dalam IAMPanduan Pengguna.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas EventBridge Penjadwal. Edit peran layanan hanya jika EventBridge Scheduler memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Scheduler EventBridge

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. AWS layanan Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan, lihat AWS layanan yang berfungsi dengannya. IAM Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Menggunakan kebijakan berbasis identitas di Scheduler EventBridge

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya EventBridge Penjadwal. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan IAM berbasis identitas menggunakan contoh dokumen kebijakan ini, lihat Membuat JSON IAM kebijakan di Panduan Pengguna. IAM

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh EventBridge Penjadwal, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk EventBridge Penjadwal Amazon di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan

- EventBridge Izin penjadwal
- AWS kebijakan terkelola untuk EventBridge Scheduler
- Kebijakan terkelola pelanggan untuk EventBridge Scheduler
- AWS pembaruan kebijakan terkelola

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber EventBridge Penjadwal di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan AWSAWS terkelola atau kebijakan terkelola untuk fungsi pekerjaan di Panduan IAM Pengguna.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat Kebijakan dan izin IAM di IAM Panduan Pengguna.
- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakanSSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik AWS layanan, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat elemen IAM JSON kebijakan: Kondisi dalam Panduan IAM Pengguna.
- Gunakan IAM Access Analyzer untuk memvalidasi IAM kebijakan Anda guna memastikan izin yang aman dan fungsional — IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa IAM kebijakan () JSON dan praktik terbaik. IAM IAMAccess Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang

dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan IAM Access Analyzer di IAMPanduan Pengguna.

- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan IAM pengguna atau pengguna root di dalam Anda Akun AWS, aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA kapan API operasi dipanggil, tambahkan MFA kondisi ke kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi API akses MFA yang dilindungi di IAMPanduan Pengguna.

Untuk informasi selengkapnya tentang praktik terbaik diIAM, lihat Praktik terbaik keamanan IAM di Panduan IAM Pengguna.

EventBridge Izin penjadwal

Agar IAM prinsipal (pengguna, grup, atau peran) dapat membuat jadwal di EventBridge Penjadwal dan mengakses sumber EventBridge Penjadwal melalui konsol atauAPI, prinsipal harus memiliki serangkaian izin yang ditambahkan ke kebijakan izin mereka. Anda dapat mengonfigurasi izin ini tergantung pada fungsi pekerjaan kepala sekolah. Misalnya, pengguna, atau peran, yang hanya menggunakan konsol EventBridge Scheduler untuk melihat daftar jadwal yang ada tidak perlu memiliki izin yang diperlukan untuk memanggil operasi. `CreateSchedule` API Sebaiknya sesuaikan izin berbasis identitas Anda untuk hanya memberikan akses istimewa yang paling sedikit.

Daftar berikut menunjukkan sumber daya EventBridge Scheduler, dan tindakan yang didukung terkait.

- Jadwal
  - `scheduler:ListSchedules`
  - `scheduler:GetSchedule`
  - `scheduler:CreateSchedule`
  - `scheduler:UpdateSchedule`
  - `scheduler:DeleteSchedule`
- Jadwalkan grup
  - `scheduler:ListScheduleGroups`
  - `scheduler:GetScheduleGroup`
  - `scheduler:CreateScheduleGroup`
  - `scheduler:DeleteScheduleGroup`

- `scheduler:ListTagsForResource`
- `scheduler:TagResource`
- `scheduler:UntagResource`

Anda dapat menggunakan izin EventBridge Scheduler untuk membuat kebijakan terkelola pelanggan Anda sendiri untuk digunakan dengan EventBridge Scheduler. Anda juga dapat menggunakan kebijakan AWS terkelola yang dijelaskan di bagian berikut untuk memberikan izin yang diperlukan untuk kasus penggunaan umum tanpa harus mengelola kebijakan Anda sendiri.

AWS kebijakan terkelola untuk EventBridge Scheduler

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan IAM kebijakan mandiri yang AWS membuat, dan mengelola. Kebijakan terkelola, atau yang ditentukan sebelumnya memberikan izin yang diperlukan untuk kasus penggunaan umum sehingga Anda tidak perlu menyelidiki izin yang diperlukan. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola di Panduan IAM Pengguna. Kebijakan AWS terkelola berikut yang dapat Anda lampirkan ke pengguna di akun Anda khusus untuk EventBridge Scheduler:

- the section called “`AmazonEventBridgeSchedulerFullAccess`”— Memberikan akses penuh ke EventBridge Scheduler menggunakan konsol dan. API
- the section called “`AmazonEventBridgeSchedulerReadOnlyAccess`”— Memberikan akses hanya-baca ke Scheduler. EventBridge

`AmazonEventBridgeSchedulerFullAccess`

Kebijakan `AmazonEventBridgeSchedulerFullAccess` terkelola memberikan izin untuk menggunakan semua tindakan EventBridge Penjadwal untuk jadwal, dan grup jadwal.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "scheduler:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

`AmazonEventBridgeSchedulerReadOnlyAccess`

Kebijakan `AmazonEventBridgeSchedulerReadOnlyAccess` terkelola memberikan izin hanya-baca untuk melihat detail tentang jadwal dan grup jadwal Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "scheduler:ListSchedules",
                "scheduler:ListScheduleGroups",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan terkelola pelanggan untuk EventBridge Scheduler

Gunakan contoh berikut untuk membuat kebijakan terkelola pelanggan Anda sendiri untuk EventBridge Scheduler. Kebijakan terkelola pelanggan memungkinkan Anda memberikan izin hanya untuk tindakan dan sumber daya yang diperlukan untuk aplikasi dan pengguna di tim Anda sesuai dengan fungsi pekerjaan kepala sekolah.

Topik

- Contoh: CreateSchedule

- Contoh: GetSchedule
- Contoh: UpdateSchedule
- Contoh: DeleteScheduleGroup

Contoh: **`CreateSchedule`**

Ketika Anda membuat jadwal baru, Anda memilih apakah akan mengenkripsi data Anda di EventBridge Scheduler menggunakan Kunci milik AWS, atau kunci yang dikelola pelanggan.

Kebijakan berikut memungkinkan kepala sekolah untuk membuat jadwal dan menerapkan enkripsi menggunakan file Kunci milik AWS. Dengan Kunci milik AWS, AWS mengelola sumber daya on AWS Key Management Service (AWS KMS) untuk Anda sehingga Anda tidak memerlukan izin tambahan untuk berinteraksi AWS KMS.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:CreateSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
```

```
}
```

Gunakan kebijakan berikut untuk mengizinkan prinsipal membuat jadwal dan menggunakan kunci yang dikelola AWS KMS pelanggan untuk enkripsi. Untuk menggunakan kunci yang dikelola pelanggan, kepala sekolah harus memiliki izin untuk mengakses AWS KMS sumber daya di akun Anda. Kebijakan ini memberikan akses ke satu KMS kunci tertentu yang akan digunakan untuk mengenkripsi data pada EventBridge Scheduler. Atau, Anda dapat menggunakan karakter wildcard (*) untuk memberikan akses ke semua kunci di akun, atau subset yang cocok dengan pola nama tertentu.

```
{
    "Version": "2012-10-17"
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:CreateSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        },
        {
            "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:kms:us-west-2:123456789012:key/my-key-id"
            ],
            "Conditions": {
                "StringLike": {
                    "kms:ViaService": "scheduler.amazonaws.com",
```

```
                    "kms:EncryptionContext:aws:scheduler:schedule:arn":
 "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
            }
        }
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

Contoh: **`GetSchedule`**

Gunakan kebijakan berikut untuk memungkinkan kepala sekolah mendapatkan informasi tentang jadwal.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:GetSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        }
    ]
}
```

Contoh: **`UpdateSchedule`**

Gunakan kebijakan berikut untuk mengizinkan prinsipal memperbarui jadwal dengan memanggil `scheduler:UpdateSchedule` tindakan. Mirip dengan`CreateSchedule`, kebijakan tergantung pada apakah jadwal menggunakan AWS KMS Kunci milik AWS atau kunci yang dikelola pelanggan untuk enkripsi. Untuk jadwal yang dikonfigurasi dengan Kunci milik AWS, gunakan kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:UpdateSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

Untuk jadwal yang dikonfigurasi dengan kunci terkelola pelanggan, gunakan kebijakan berikut. Kebijakan ini mencakup izin tambahan yang memungkinkan prinsipal mengakses AWS KMS sumber daya di akun Anda:

```
{
    "Version": "2012-10-17",
```

```
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:UpdateSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name
        },
        {
            "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:kms:us-west-2:123456789012:key/my-key-id"
            ],
            "Conditions": {
                "StringLike": {
                    "kms:ViaService": "scheduler.amazonaws.com",
                    "kms:EncryptionContext:aws:scheduler:schedule:arn":
 "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
            }
        }
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
```

```
}
```

Contoh: **`DeleteScheduleGroup`**

Gunakan kebijakan berikut untuk mengizinkan kepala sekolah menghapus grup jadwal. Saat menghapus grup, Anda juga menghapus jadwal yang terkait dengan grup tersebut. Prinsipal yang menghapus grup harus memiliki izin untuk juga menghapus jadwal yang terkait dengan grup itu. Kebijakan ini memberikan izin utama untuk memanggil `scheduler:DeleteScheduleGroup` tindakan pada kelompok jadwal yang ditentukan, serta semua jadwal dalam grup:

> **Note**
>
> EventBridge Scheduler tidak mendukung menentukan izin tingkat sumber daya untuk jadwal individu. Misalnya, pernyataan berikut tidak valid dan tidak boleh disertakan dalam kebijakan Anda:
> `"Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "scheduler:DeleteSchedule",
            "Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/*"
        },
        {
            "Effect": "Allow",
            "Action": "scheduler:DeleteScheduleGroup",
            "Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
```

```
        }
    ]
}
```

AWS pembaruan kebijakan terkelola

Perubahan	Deskripsi	Tanggal
the section called "AmazonEventBridgeSchedulerFullAccess"— Kebijakan terkelola baru	EventBridge Scheduler menambahkan dukungan untuk kebijakan terkelola baru yang memberi pengguna akses penuh ke semua sumber daya, termasuk jadwal, dan grup jadwal.	10 November 2022
the section called "AmazonEventBridgeSchedulerReadOnlyAccess"— Kebijakan terkelola baru	EventBridge Scheduler menambahkan dukungan untuk kebijakan terkelola baru yang memberi pengguna akses hanya-baca ke semua sumber daya, termasuk jadwal, dan grup jadwal.	10 November 2022
EventBridge Scheduler mulai melacak perubahan	EventBridge Scheduler mulai melacak perubahan untuk kebijakan yang AWS dikelola.	10 November 2022

Pencegahan Deputi Bingung di EventBridge Scheduler

Masalah deputi yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah

hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan dalam peran eksekusi jadwal Anda untuk membatasi izin yang diberikan EventBridge Scheduler kepada layanan lain untuk mengakses sumber daya. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci konteks kondisi `aws:SourceArn` global dengan penuh ARN sumber daya. Kondisi berikut dicakup oleh kelompok jadwal individu: `arn:aws:scheduler:*:123456789012:schedule-group/your-schedule-group`

Jika Anda tidak tahu sumber daya penuh ARN atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (`*`) untuk bagian yang tidak diketahui dari file. ARN Sebagai contoh: `arn:aws:scheduler:*:123456789012:schedule-group/*`.

Nilai `aws:SourceArn` harus menjadi grup jadwal EventBridge Scheduler Anda yang ARN ingin Anda cakup kondisi ini.

> **Important**
>
> Jangan lingkup `aws:SourceArn` pernyataan ke jadwal tertentu atau awalan nama jadwal. Yang ARN Anda tentukan harus berupa grup jadwal.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global dalam kebijakan kepercayaan peran eksekusi Anda untuk mencegah masalah deputi yang membingungkan:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "scheduler.amazonaws.com"
```

```
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012",
                    "aws:SourceArn": "arn:aws:scheduler:us-
west-2:123456789012:schedule-group/your-schedule-group"
                }
            }
        }
    ]
}
```

Memecahkan masalah identitas dan akses Amazon EventBridge Scheduler

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan EventBridge Scheduler danIAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di EventBridge Scheduler
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya EventBridge Penjadwal saya

Saya tidak berwenang untuk melakukan tindakan di EventBridge Scheduler

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika `mateojackson` IAM pengguna mencoba menggunakan konsol untuk melihat detail tentang *`my-example-widget`* sumber daya fiksi tetapi tidak memiliki izin `scheduler:`*`GetWidget`* fiksi.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 scheduler:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan Mateo harus diperbarui untuk memungkinkannya mengakses *`my-example-widget`* sumber daya menggunakan `scheduler:`*`GetWidget`* tindakan tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke EventBridge Scheduler.

Beberapa AWS layanan memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika IAM pengguna bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di EventBridge Scheduler. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya EventBridge Penjadwal saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah EventBridge Scheduler mendukung fitur ini, lihatBagaimana EventBridge Scheduler bekerja dengan IAM.

- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke IAM pengguna lain Akun AWS yang Anda miliki di Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna yang diautentikasi secara eksternal (federasi identitas) di Panduan Pengguna. IAM
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM Panduan Pengguna. IAM

Perlindungan data di Amazon EventBridge Scheduler

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di Amazon EventBridge Scheduler. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk AWS layanan yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat Privasi Data FAQ. Untuk informasi tentang perlindungan data di Eropa, lihat Model Tanggung Jawab AWS Bersama dan posting GDPR blog di Blog AWS Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management ()IAM. Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- GunakanSSL/TLSuntuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya AWS layanan.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.

- Jika Anda memerlukan FIPS 140-3 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atau, gunakan titik akhir. API FIPS Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat Federal Information Processing Standard (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk ketika Anda bekerja dengan EventBridge Scheduler atau lainnya AWS layanan menggunakan konsol,, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Jika Anda memberikan URL ke server eksternal, kami sangat menyarankan agar Anda tidak menyertakan informasi kredensil dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi saat istirahat di EventBridge Scheduler

Bagian ini menjelaskan cara Amazon EventBridge Scheduler mengenkripsi dan mendekripsi data Anda saat istirahat. Data saat istirahat adalah data yang disimpan dalam EventBridge Scheduler dan komponen dasar layanan. EventBridge Scheduler terintegrasi dengan AWS Key Management Service (AWS KMS) untuk mengenkripsi dan mendekripsi data Anda menggunakan file. AWS KMS key EventBridge Scheduler mendukung dua jenis KMS kunci: Kunci milik AWS, dan kunci yang dikelola pelanggan.

> **Note**
>
> EventBridge Scheduler hanya mendukung penggunaan kunci enkripsi KMSsimetris.

Kunci milik AWSadalah KMS kunci yang dimiliki dan dikelola AWS layanan untuk digunakan di beberapa AWS akun. Meskipun penggunaan Kunci milik AWS EventBridge Scheduler tidak disimpan di AWS akun Anda, EventBridge Scheduler menggunakannya untuk melindungi data dan sumber daya Anda. Secara default, EventBridge Scheduler mengenkripsi dan mendekripsi semua data Anda menggunakan kunci yang dimiliki. AWS Anda tidak perlu mengelola kebijakan akses Anda Kunci milik AWS atau nya. Anda tidak dikenakan biaya apa pun ketika EventBridge Scheduler menggunakan Kunci milik AWS untuk melindungi data Anda, dan penggunaannya tidak dihitung sebagai bagian dari AWS KMS kuota Anda di akun Anda.

Kunci yang dikelola pelanggan adalah KMS kunci yang disimpan di AWS akun Anda yang Anda buat, miliki, dan kelola. Jika kasus penggunaan khusus Anda mengharuskan Anda mengontrol

dan mengaudit kunci enkripsi yang melindungi data Anda di EventBridge Scheduler, Anda dapat menggunakan kunci yang dikelola pelanggan. Jika Anda memilih kunci yang dikelola pelanggan, Anda harus mengelola kebijakan utama Anda. Kunci yang dikelola pelanggan dikenakan biaya bulanan dan biaya untuk penggunaan melebihi tingkat gratis. Menggunakan kunci yang dikelola pelanggan juga dihitung sebagai bagian dari AWS KMS kuota Anda. Untuk informasi lebih lanjut tentang harga, lihat AWS Key Management Service harga.

Topik

Artefak enkripsi

Tabel berikut menjelaskan berbagai jenis data yang dienkripsi EventBridge Scheduler saat istirahat, dan jenis KMS kunci yang didukungnya untuk setiap kategori.

Tipe data	Deskripsi	Kunci milik AWS	kunci yang dikelola pelanggan
Muatan (hingga 256KB)	Data yang Anda tentukan dalam `TargetInput` parameter jadwal saat Anda mengonfigurasi jadwal yang akan dikirim ke target.	Didukung	Didukung
Pengenal dan status	Nama unik dan status (aktifkan, nonaktifkan) dari jadwal.	Didukung	Tidak Support
Konfigurasi penjadwalan	Ekspresi penjadwalan, seperti ekspresi rate atau cron untuk jadwal berulang, dan stempel waktu untuk	Didukung	Tidak Support

Tipe data	Deskripsi	Kunci milik AWS	kunci yang dikelola pelanggan
	pemanggilan satu kali, serta tanggal mulai jadwal, tanggal akhir, dan zona waktu.		
Konfigurasi target	Nama Sumber Daya Amazon (ARN) target, dan detail konfigurasi terkait target lainnya.	Didukung	Tidak Support
Konfigurasi perilaku pemanggilan dan kegagalan	Konfigurasi jendela waktu yang fleksibel, kebijakan coba ulang jadwal, dan detail antrian surat mati yang digunakan untuk pengiriman yang gagal.	Didukung	Tidak Support

EventBridge Scheduler hanya menggunakan kunci terkelola pelanggan Anda saat mengenkripsi dan mendekripsi muatan target, seperti yang dijelaskan dalam tabel sebelumnya. Jika Anda memilih untuk menggunakan kunci yang dikelola pelanggan, EventBridge Scheduler mengenkripsi dan mendekripsi payload dua kali: sekali menggunakan default Kunci milik AWS, dan lain kali menggunakan kunci terkelola pelanggan yang Anda tentukan. Untuk semua tipe data lainnya, EventBridge Scheduler hanya menggunakan default Kunci milik AWS untuk melindungi data Anda saat istirahat.

Gunakan the section called “Mengelola KMS kunci” bagian berikut untuk mempelajari cara mengelola IAM sumber daya dan kebijakan utama agar dapat menggunakan kunci yang dikelola pelanggan dengan EventBridge Scheduler.

Mengelola KMS kunci

Anda dapat secara opsional memberikan kunci yang dikelola pelanggan untuk mengenkripsi dan mendekripsi muatan yang dikirim jadwal Anda ke targetnya. EventBridge Scheduler mengenkripsi

dan mendekripsi payload Anda hingga 256KB data. Menggunakan kunci yang dikelola pelanggan menimbulkan biaya bulanan dan biaya melebihi tingkat gratis. Menggunakan kunci yang dikelola pelanggan dihitung sebagai bagian dari AWS KMS kuota Anda. Untuk informasi lebih lanjut tentang harga, lihat AWS Key Management Service harga

EventBridge Scheduler menggunakan IAM izin yang terkait dengan prinsipal yang membuat jadwal untuk mengenkripsi data Anda. Ini berarti Anda harus melampirkan izin AWS KMS terkait yang diperlukan ke pengguna, atau peran, yang memanggil EventBridge PenjadwalAPI. Selain itu, EventBridge Scheduler menggunakan kebijakan berbasis sumber daya untuk mendekripsi data Anda. Ini berarti bahwa peran eksekusi yang terkait dengan jadwal Anda juga harus memiliki izin AWS KMS terkait yang diperlukan untuk memanggil AWS KMS API saat mendekripsi data.

> **Note**
>
> EventBridge Scheduler tidak mendukung penggunaan hibah untuk izin sementara.

Gunakan bagian berikut untuk mempelajari cara mengelola kebijakan AWS KMS utama dan IAM izin yang diperlukan untuk menggunakan kunci yang dikelola pelanggan di EventBridge Scheduler.

Topik

- Tambahkan IAM izin
- Mengelola kebijakan utama

Tambahkan IAM izin

Untuk menggunakan kunci terkelola pelanggan, Anda harus menambahkan izin berikut ke IAM prinsipal berbasis identitas yang membuat jadwal, serta peran eksekusi yang Anda kaitkan dengan jadwal.

Izin berbasis identitas untuk kunci terkelola pelanggan

Anda harus menambahkan AWS KMS tindakan berikut ke kebijakan izin yang terkait dengan prinsipal (pengguna, grup, atau peran) yang memanggil EventBridge Penjadwal API saat membuat jadwal.

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

```
        {
            "Action": [
                "scheduler:*",

                # Required to pass the execution role
                "iam:PassRole",

                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
    ]
}
```

- **`kms:DescribeKey`**— Diperlukan untuk memvalidasi bahwa kunci yang Anda berikan adalah kunci enkripsi KMSsimetris.
- **`kms:GenerateDataKey`**— Diperlukan untuk menghasilkan kunci data yang digunakan EventBridge Scheduler untuk melakukan enkripsi sisi klien.
- **`kms:Decrypt`**— Diperlukan dekripsi kunci data terenkripsi yang disimpan EventBridge Scheduler bersama dengan data terenkripsi Anda.

Izin peran eksekusi untuk kunci terkelola pelanggan

Anda harus menambahkan tindakan berikut ke kebijakan izin peran eksekusi jadwal Anda untuk menyediakan akses ke EventBridge Scheduler untuk memanggil AWS KMS API saat mendekripsi data Anda.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
      "Sid" : "Allow EventBridge Scheduler to decrypt data using a customer managed
 key",
      "Effect" : "Allow",
      "Action" : [
        "kms:Decrypt"

      ],
```

```
      "Resource": "arn:aws:kms:your-region:123456789012:key/your-key-id"
    }
  ]
}
```

- **`kms:Decrypt`**— Diperlukan dekripsi kunci data terenkripsi yang disimpan EventBridge Scheduler bersama dengan data terenkripsi Anda.

Jika Anda menggunakan konsol EventBridge Scheduler untuk membuat peran eksekusi baru saat membuat jadwal baru, EventBridge Scheduler akan secara otomatis melampirkan izin yang diperlukan ke peran eksekusi Anda. Namun, jika Anda memilih peran eksekusi yang ada, Anda harus menambahkan izin yang diperlukan ke peran tersebut agar dapat menggunakan kunci terkelola pelanggan Anda.

Mengelola kebijakan utama

Saat Anda membuat kunci terkelola pelanggan menggunakan AWS KMS, secara default, kunci Anda memiliki kebijakan kunci berikut untuk menyediakan akses ke peran eksekusi jadwal Anda.

```
{
    "Id": "key-policy-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Provide required IAM Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

Secara opsional, Anda dapat membatasi cakupan kebijakan utama Anda untuk hanya menyediakan akses ke peran eksekusi. Anda dapat melakukan ini jika Anda ingin menggunakan kunci terkelola pelanggan Anda hanya dengan sumber EventBridge Scheduler Anda. Gunakan contoh kebijakan kunci berikut untuk membatasi sumber daya EventBridge Scheduler mana yang dapat menggunakan kunci Anda.

```
{
    "Id": "key-policy-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Provide required IAM Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::695325144837:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/schedule-execution-role"
            },
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudTrail contoh acara

AWS CloudTrail menangkap semua acara API panggilan. Ini termasuk API panggilan setiap kali EventBridge Scheduler menggunakan kunci yang dikelola pelanggan Anda untuk mendekripsi data Anda. Contoh berikut menunjukkan entri CloudTrail peristiwa yang menunjukkan EventBridge Scheduler menggunakan `kms:Decrypt` tindakan menggunakan kunci yang dikelola pelanggan.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
      "type": "AssumedRole",
      "principalId": "ABCDEABCD1AB12ABABAB0:70abcd123a123a12345a1aa12aa1bc12",
      "arn": "arn:aws:sts::123456789012:assumed-role/execution-
role/70abcd123a123a12345a1aa12aa1bc12",
      "accountId": "123456789012",
```

```
      "accessKeyId": "ABCDEFGHI1JKLMNOP2Q3",
      "sessionContext": {
          "sessionIssuer": {
              "type": "Role",
              "principalId": "ABCDEABCD1AB12ABABAB0",
              "arn": "arn:aws:iam::123456789012:role/execution-role",
              "accountId": "123456789012",
              "userName": "execution-role"
          },
          "webIdFederationData": {},
          "attributes": {
              "creationDate": "2022-10-31T21:03:15Z",
              "mfaAuthenticated": "false"
          }
      }
    },
    "eventTime": "2022-10-31T21:03:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-north-1",
    "sourceIPAddress": "13.50.87.173",
    "userAgent": "aws-sdk-java/2.17.295 Linux/4.14.291-218.527.amzn2.x86_64 OpenJDK_64-
Bit_Server_VM/11.0.17+9-LTS Java/11.0.17 kotlin/1.3.72-release-468 (1.3.72) vendor/
Amazon.com_Inc. md/internal exec-env/AWS_ECS_FARGATE io/sync http/Apache cfg/retry-
mode/standard AwsCrypto/2.4.0",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-west-2:123456789012:key/2321abab-2110-12ab-a123-
a2b34c5abc67",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "aws:scheduler:schedule:arn": "arn:aws:scheduler:us-
west-2:123456789012:schedule/default/execution-role"
        }
    },
    "responseElements": null,
    "requestID": "request-id",
    "eventID": "event-id",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:123456789012:key/2321abab-2110-12ab-a123-
a2b34c5abc67"
```

```
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_256_GCM_SHA384",
      "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
  }
}
```

Enkripsi dalam perjalanan di EventBridge Scheduler

EventBridge Scheduler mengenkripsi data Anda dalam perjalanan saat melakukan perjalanan jaringan. Transport Layer Security (TLS) mengenkripsi data Anda saat Anda memanggil API operasi EventBridge Scheduler apa pun, serta saat EventBridge Scheduler memanggil target apa pun APIs saat memanggil jadwal Anda. Secara default, EventBridge Scheduler menggunakan TLS 1.2 saat mengenkripsi data Anda dalam perjalanan. Anda tidak perlu mengonfigurasi enkripsi saat transit, dan Anda tidak dapat memilih TLS versi yang berbeda saat menggunakan EventBridge Scheduler.

Menggunakan EventBridge Scheduler API — Saat Anda melakukan API operasi, sepertiCreateSchedule, EventBridge Scheduler mengenkripsi seluruh HTTP permintaan, termasuk badan permintaan dan header. EventBridge Scheduler juga mengenkripsi seluruh objek respons yang Anda terima dari kami. APIs

Menggunakan target APIs — Ketika EventBridge Scheduler memanggil jadwal Anda, itu memanggil target API yang Anda tentukan saat Anda membuat jadwal. Saat mengirimkan acara ke target, EventBridge Scheduler mengenkripsi seluruh permintaan, termasuk badan permintaan dan semua header, serta respons yang diterimanya dari target.

Validasi kepatuhan untuk Amazon Scheduler EventBridge

Untuk mempelajari apakah an AWS layanan berada dalam lingkup program kepatuhan tertentu, lihat AWS layanan di Lingkup oleh Program Kepatuhan AWS layanan dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan AWS layanan ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- Arsitektur untuk HIPAA Keamanan dan Kepatuhan di Amazon Web Services — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat HIPAA aplikasi yang memenuhi syarat.

> **Note**
>
> Tidak semua AWS layanan HIPAA memenuhi syarat. Untuk informasi selengkapnya, lihat Referensi Layanan yang HIPAA Memenuhi Syarat.

- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan AWS layanan dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi ()). ISO
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini AWS layanan memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty — Ini AWS layanan mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCIDSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.

- AWS Audit ManagerIni AWS layanan membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan di Amazon Scheduler EventBridge

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Selain infrastruktur AWS global, EventBridge Scheduler menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

Keamanan Infrastruktur di Amazon EventBridge Scheduler

Sebagai layanan terkelola, Amazon EventBridge Scheduler dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan API panggilan yang AWS dipublikasikan untuk mengakses EventBridge Scheduler melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Transportasi (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju yang sempurna (PFS) seperti (Ephemeral Diffie-Hellman) atau DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Atau Anda bisa menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Pemantauan dan metrik untuk Amazon Scheduler EventBridge

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon EventBridge Scheduler dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton EventBridge Scheduler, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- AWS CloudTrailmenangkap API panggilan dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.

Topik

Memantau EventBridge Penjadwal Amazon dengan Amazon CloudWatch

Anda dapat memantau Amazon EventBridge Scheduler menggunakan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. EventBridge Scheduler memancarkan satu set metrik untuk semua jadwal, dan satu set metrik tambahan untuk jadwal yang memiliki antrian huruf mati terkait (). DLQ Jika Anda mengonfigurasi jadwal, EventBridge Scheduler akan menerbitkan metrik tambahan saat jadwal Anda habis kebijakan coba ulangnya. DLQ

Statistik ini disimpan selama 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang mengapa jadwal gagal, dan memecahkan masalah mendasar. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

Topik

Ketentuan

Namespace

Namespace adalah wadah untuk CloudWatch metrik layanan. AWS Untuk EventBridge Scheduler, namespace adalah. `AWS/Scheduler`

CloudWatch metrik

CloudWatch Metrik mewakili kumpulan titik data yang diurutkan waktu yang spesifik untuk CloudWatch.

Dimensi

Dimensi adalah pasangan nama/nilai yang merupakan bagian dari identitas metrik.

Unit

Statistik memiliki satuan ukuran. Untuk EventBridge Scheduler, unit termasuk Count.

Dimensi

Bagian ini menjelaskan pengelompokan CloudWatch dimensi untuk metrik EventBridge Scheduler di. CloudWatch

Dimensi	Deskripsi
ScheduleGroup	Kelompok jadwal yang ingin Anda lihat metrik menggunakan. CloudWatch Jika Anda belum membuat grup apa pun, EventBridge Scheduler mengaitkan jadwal Anda dengan grup. `default`

Mengakses metrik

Bagian ini menjelaskan cara mengakses metrik kinerja CloudWatch untuk EventBridge jadwal Penjadwal tertentu.

Untuk melihat metrik kinerja untuk dimensi

1. Buka halaman Metrik di CloudWatch konsol.
2. Gunakan pemilih AWS Wilayah untuk memilih Wilayah untuk jadwal Anda
3. Pilih namespace Scheduler.
4. Di tab Semua metrik, pilih dimensi, misalnya, Jadwalkan Metrik Grup. Untuk melihat metrik untuk semua jadwal yang telah Anda buat di Wilayah yang dipilih, pilih Metrik Akun.
5. Pilih CloudWatch metrik untuk dimensi. Misalnya, InvocationAttemptCountatau InvocationDroppedCount, lalu pilih Pencarian grafik.
6. Pilih tab Graphed metrics untuk melihat statistik performa untuk metrik EventBridge Scheduler.

Daftar metrik

Tabel berikut mencantumkan metrik untuk semua EventBridge jadwal Scheduler, serta metrik tambahan untuk jadwal yang telah Anda konfigurasi. DLQ

Metrik untuk semua jadwal

Namespace	Metrik	Unit	Deskripsi
`AWS/Scheduler`	`InvocationAttemptCount`	Hitung	Dipancarkan untuk setiap upaya doa.

Namespace	Metrik	Unit	Deskripsi
			Gunakan metrik ini untuk memeriksa apakah EventBridge Scheduler mencoba memanggil jadwal Anda, dan untuk melihat kapan pemanggilan mendekati kuota akun Anda.
`AWS/Scheduler`	`TargetErrorCount`	Hitung	Dipancarkan ketika target mengembalikan pengecualian setelah EventBridge Scheduler memanggil target. API Gunakan ini untuk memeriksa kapan pengiriman ke target gagal.

Namespace	Metrik	Unit	Deskripsi
AWS/Scheduler	TargetErrorThrottledCount	Hitung	Dipancarkan saat pemanggilan target gagal karena pembatasan oleh target. API Gunakan ini untuk mendiagnosis kegagalan pengiriman ketika alasan yang mendasarinya adalah panggilan API pembatasan target yang dilakukan oleh Scheduler EventBridge
AWS/Scheduler	InvocationThrottleCount	Hitung	Dipancarkan saat EventBridge Scheduler membatasi pemanggilan target karena melebihi kuota layanan Anda yang ditetapkan oleh Scheduler. EventBridge Gunakan ini untuk menentukan kapan Anda telah melampaui kuota batas throttle pemanggilan Anda. Untuk informasi lebih lanjut tentang kuota layanan, lihatKuota.

Namespace	Metrik	Unit	Deskripsi
`AWS/Scheduler`	`InvocationDroppedCount`	Hitung	Dipancarkan ketika EventBridge Scheduler berhenti mencoba untuk memanggil target setelah kebijakan coba ulang jadwal telah habis. Untuk informasi selengkapnya tentang kebijakan coba lagi, lihat RetryPolicydi Referensi EventBridge Penjadwal API.

Metrik untuk jadwal dengan a DLQ

Namespace	Metrik	Unit	Deskripsi
`AWS/Scheduler`	`InvocationsSentToDeadLetterCount`	Hitung	Dipancarkan untuk setiap pengiriman yang berhasil ke jadwal. DLQ Gunakan ini untuk menentukan kapan acara dikirim ke aDLQ, lalu periksa acara yang dikirimkan ke

Namespace	Metrik	Unit	Deskripsi
			jadwal DLQ untuk detail tambahan yang membantu Anda menentukan penyebab kegagalan.

Namespace	Metrik	Unit	Deskripsi
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount	Hitung	Dipancarkan ketika EventBridge Scheduler tidak dapat mengirimkan acara ke. DLQ Gunakan dua metrik ini untuk menentukan alasan mengapa EventBridge Scheduler tidak dapat mengirim acara keDLQ, dan memodifikasi DLQ konfigurasi Anda untuk menyelesaikan masalah. Berikut ini adalah contoh InvocationsFailedToBeSentToDeadLetterCount_<e
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount_<error_code>	Hitung	

Namespace	Metrik	Unit	Deskripsi
			`rror_code>` metrik saat SQS antrian Amazon yang Anda tentukan sebagai DLQ tidak ada: `InvocationsFailedToBeSentToDeadLetterCount_` **`AWS.SimpleQueueService.NonExistentQueue`**

Namespace	Metrik	Unit	Deskripsi
`AWS/Scheduler`	`InvocationsSentToDeadLetterCount_Truncated_MessageSizeExceeded`	Hitung	Dipancarkan ketika muatan acara yang dikirim ke DLQ melebihi ukuran maksimum yang diizinkan oleh AmazonSQS, dan EventBridge Scheduler memotong payload yang Anda tentukan dalam atribut jadwal. `Input`

EventBridge Metrik penggunaan penjadwal

CloudWatch mengumpulkan metrik yang melacak penggunaan beberapa AWS sumber daya. Metrik ini sesuai dengan kuota AWS layanan. Dengan melacak metrik-metrik tersebut dapat membantu Anda mengelola kuota secara proaktif. Gunakan metrik berikut untuk menentukan kapan Anda telah melampaui kuota EventBridge Scheduler Anda. Untuk informasi lebih lanjut tentang kuota layanan, lihatKuota.

Metrik ini terkandung dalam `AWS/Usage` namespace, bukan`AWS/Scheduler`, dan dikumpulkan setiap menit.

Saat ini, satu-satunya nama metrik di namespace ini yang CloudWatch diterbitkan adalah. `CallCount` Metrik ini diterbitkan dengan dimensi `Resource`, `Service`, dan `Type`. `Resource`Dimensi menentukan nama API operasi yang dilacak.

Misalnya, `CallCount` metrik dengan dimensi berikut menunjukkan berapa kali Penjadwal EventBridge `CreateSchedule` API operasi telah dipanggil di akun Anda:

- “Layanan”: “Penjadwal”
- “Ketik”: "API”
- “Sumber”: "CreateSchedule”

Metrik `CallCount` tidak memiliki unit tertentu. Statistik yang paling berguna untuk metrik adalah `SUM`, yang menunjukkan total operasi untuk periode 1 menit.

Metrik-metrik

Metrik	Deskripsi		
`CallCount`	Jumlah operasi tertentu yang dilakukan di akun Anda.		

Dimensi

Dimensi	Deskripsi		
`Service`	Nama AWS layanan yang berisi sumber daya. Untuk metrik Penjadwal EventBridge penggunaan, nilai untuk dimensi ini adalah`Scheduler` .		
`Class`	Kelas sumber daya yang akan dilacak.		

Dimensi	Deskripsi		
	Penjadwal EventBridge APImetrik penggunaan menggunakan dimensi ini dengan nilai. `None`		
`Type`	Jenis sumber daya yang sedang ditelusuri. Saat ini, ketika dimensi `Service` adalah `Scheduler` , satu-satunya nilai yang benar untuk `Type` adalah `API`.		
`Resource`	Nama API operasi. Nilai-nilai yang valid meliputi: • `CreateSchedule` • `CreateScheduleGroup` • `DeleteSchedule` • `DeleteScheduleGroup` • `GetSchedule` • `GetScheduleGroup` • `ListScheduleGroups` • `ListSchedules` • `ListTagsForResource` • `TagResource` • `UntagResource` • `UpdateSchedule`		

Pencatatan API panggilan Amazon EventBridge Scheduler menggunakan AWS CloudTrail

Amazon EventBridge Scheduler terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di EventBridge Scheduler. CloudTrail menangkap semua API panggilan untuk EventBridge Scheduler sebagai acara. Panggilan yang diambil termasuk panggilan dari konsol EventBridge Scheduler dan panggilan kode ke operasi EventBridge SchedulerAPI. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk EventBridge Scheduler. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk EventBridge Scheduler, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

EventBridge Informasi penjadwal di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di EventBridge Scheduler, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk EventBridge Scheduler, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengkonfigurasi SNS notifikasi Amazon untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua API tindakan EventBridge Penjadwal dicatat oleh CloudTrail dan didokumentasikan dalam Referensi EventBridge Penjadwal API Amazon. Misalnya, panggilan ke`CreateSchedule`, `UpdateSchedule` dan `DeleteSchedule` tindakan menghasilkan entri dalam file CloudTrail log.

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

- Apakah permintaan dibuat dengan root atau AWS Identity and Access Management (IAM) kredensional pengguna.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lebih lanjut, lihat CloudTrail userIdentityelemen.

Memahami EventBridge entri file log Scheduler

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

Kuota untuk Amazon Scheduler EventBridge

AWS Akun Anda memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah. Anda dapat meminta kenaikan untuk sebagian besar kuota, tetapi beberapa tidak dapat ditingkatkan.

Untuk melihat kuota EventBridge Scheduler, buka konsol Service Quotas. Di panel navigasi, pilih AWS layanan, lalu pilih EventBridge Scheduler.

Untuk meminta penambahan kuota, lihat Meminta penambahan kuota di Panduan Pengguna Service Quotas. Jika kuota belum tersedia dalam Service Quotas, gunakan formulir penambahan batas.

AWS Akun Anda memiliki kuota berikut yang terkait dengan EventBridge Scheduler.

Nama	Default	Dapat disesu an	Deskripsi
CreateSchedule tingkat permintaan	ca-central-1:250 eu-central-1:1.000 Masing-masing Wilayah yang didukung lainnya: 50	Ya	CreateSchedule Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
CreateScheduleGroup tingkat permintaa n	Setiap Wilayah yang didukung: 10	Ya	CreateScheduleGroup Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
DeleteSchedule tingkat permintaan	ca-central-1:250	Ya	DeleteSchedule Permintaan maksimum

Nama	Default	Dapat disesu an	Deskripsi
	eu-central-1:1.000 Masing-masing Wilayah yang didukung lainnya: 50		per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
DeleteScheduleGroup tingkat permintaa n	Setiap Wilayah yang didukung: 10	Ya	DeleteScheduleGroup Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
GetSchedule tingkat permintaan	ca-central-1:250 eu-central-1:1.000 Masing-masing Wilayah yang didukung lainnya: 50	Ya	GetSchedule Permintaa n maksimum per detik. Ketika Anda mencapai kuota ini, EventBrid ge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
GetScheduleGroup tingkat permintaan	Setiap Wilayah yang didukung: 10	Ya	GetScheduleGroup Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.

Nama	Default	Dapat disesuaian	Deskripsi
Batas throttle pemanggilan dalam transaksi per detik	eu-central-1:1.000 Masing-masing Wilayah yang didukung lainnya: 500	Ya	Doa adalah payload jadwal yang dikirim ke target yang ditentukan. Setelah batas tercapai, pemanggilan dibatasi; yaitu, mereka masih terjadi tetapi mereka tertunda.
ListScheduleGroups tingkat permintaan	Setiap Wilayah yang didukung: 10	Ya	ListScheduleGroups Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
ListSchedules tingkat permintaan	Setiap Wilayah yang didukung: 50	Ya	ListSchedules Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
ListTagsForResource tingkat permintaan	Setiap Wilayah yang didukung: 10	Ya	Daftar tag yang terkait dengan sumber Scheduler.
Jumlah grup jadwal	Setiap Wilayah yang didukung: 500	Ya	Jumlah maksimum grup jadwal per wilayah.

Nama	Default	Dapat disesu an	Deskripsi
Jumlah jadwal	ca-central-1:10.000.000 eu-central-1:10.000.000 Masing-masing Wilayah yang didukung lainnya: 1.000.000	Ya	Jumlah maksimum jadwal per wilayah. Kuota ini termasuk jadwal satu kali yang telah selesai berjalan. Sebaiknya konfigurasi jadwal Anda untuk dihapus secara otomatis setelah selesai menggunakan fitur ini ActionAfterCompletion .
TagResource tingkat permintaan	Setiap Wilayah yang didukung: 1	Ya	Menetapkan satu atau beberapa tag (pasangan kunci-nilai) ke sumber Scheduler yang ditentukan.
UntagResource tingkat permintaan	Setiap Wilayah yang didukung: 1	Ya	Menghapus satu atau beberapa tag dari sumber Scheduler yang ditentukan.
UpdateSchedule tingkat permintaan	ca-central-1:250 eu-central-1:1.000 Masing-masing Wilayah yang didukung lainnya: 50	Ya	UpdateSchedule Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.

Untuk informasi selengkapnya tentang kuota dan titik akhir layanan untuk EventBridge Scheduler, lihat titik akhir dan kuota Amazon EventBridge Scheduler di panduan Referensi Umum.AWS

Memecahkan masalah kuota di Scheduler EventBridge

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui terkait kuota EventBridge Scheduler.

ServiceQuotaExceededException

Saya menerima kesalahan pelambatan pada`CreateSchedule`,, `DeleteScheduleGetSchedule`, atau tingkat `UpdateSchedule` permintaan, meskipun saya di bawah batas tarif default.

Penyebab umum

Pada 7 September 2023, EventBridge Scheduler mulai mendukung (Nama Sumber Daya ScheduleGroup ARN Amazon) alih-alih kebijakan kepercayaan peran Jadwal ARN dalam pelaksanaan. Pelanggan yang diizinkan untuk terus menggunakan Jadwal ARNs dalam kebijakan kepercayaan mereka mungkin memiliki batas 50TPS, bukan batas default 250 hingga 1000 TPS (tergantung pada wilayah).

Resolusi

Hubungi dukungan untuk meminta batas maksimum yang lebih tinggi.

Pencegahan

Ubah kebijakan kepercayaan Anda yang ada dengan salah satu cara berikut:

- Menghapus semua pelingkupan dari peran.
- Pelingkupan peran sehingga dapat diasumsikan menggunakan Jadwal ARN atau. ScheduleGroup ARN

  Misalnya, Anda memiliki kebijakan kepercayaan yang ada berikut:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "scheduler.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringEquals": {
```

```
            "aws:SourceArn":
 "arn:aws:scheduler:region:account:schedule/schedule_group/schedule"
        }
    }
}
```

Anda dapat memperbarui kebijakan kepercayaan menjadi berikut:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "scheduler.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:SourceArn": [
                "arn:aws:scheduler:region:account:schedule/schedule_group/schedule",
                "arn:aws:scheduler:region:account:schedule-group/schedule_group"
            ]
        }
    }
}
```

Riwayat dokumen untuk Panduan Pengguna EventBridge Penjadwal

Tabel berikut menjelaskan rilis dokumentasi untuk EventBridge Scheduler.

Perubahan	Deskripsi	Tanggal
Perubahan peran eksekusi dan pencegahan wakil yang membingungkan	Pemutakhiran ini menjelaskan perubahan pada cara peran eksekusi diterapkan ke sumber daya grup jadwal saat Anda menerapkan pencegahan wakil yang membingungkan dalam kebijakan izin peran. • the section called “Pencegahan Deputi Bingung”	7 September 2023
Penghapusan jadwal secara otomatis setelah selesai	EventBridge Scheduler mendukung penghapusan otomatis. Saat Anda mengonfigurasi penghapusan otomatis, EventBridge Scheduler menghapus jadwal Anda setelah pemanggilan terakhir yang direncanakan. • the section called “Penghapusan setelah jadwal selesai”	2 Agustus 2023
Topik terbaru tentang penggunaan target universal	Memperbarui daftar layanan yang didukung yang dapat ditargetkan dan diintegrasikan oleh EventBridge Scheduler. Pembaruan ini	Maret 17, 2023

	juga mencakup daftar `GET` API operasi yang tidak didukung, dan mencakup peningkatan pada contoh target universal, serta perbaikan kecil lainnya di seluruh panduan. • the section called "Menggunakan target universal"	
Informasi terbaru tentang jadwal berbasis tarif yang tidak memiliki tanggal mulai	Menambahkan informasi tentang bagaimana EventBridge Scheduler menangani jadwal berbasis tarif jika Anda tidak menentukan. `StartDate` • the section called "Jadwal berbasis tarif"	Maret 17, 2023
Topik baru tentang mengelola grup penjadwal	Menambahkan babak baru tentang cara membuat grup penjadwal dengan EventBridge Scheduler. Gunakan Bab ini untuk mempelajari cara membuat grup, menambahkan jadwal ke grup, menerapkan tag untuk mengelola dan memonirot sumber daya EventBridge Scheduler Anda dengan lebih mudah, dan akhirnya menghapus grup. • Mengelola grup jadwal	Maret 17, 2023

Topik baru tentang waktu musim panas dan zona waktu	Menambahkan bagian baru yang menjelaskan bagaimana EventBridge Scheduler menangani daylight saving time, dan bagaimana Anda dapat membuat jadwal di zona waktu yang berbeda. • the section called “Waktu penghematan siang hari” • the section called “Zona waktu”	17 November 2022
Topik baru tentang metrik	Menambahkan topik baru yang menjelaskan metrik yang diterbitkan oleh EventBridge Scheduler. CloudWatch Anda dapat menggunakan metrik ini untuk memantau kegagalan pemanggilan dan memahami cara menyelesaikan masalah dengan jadwal Anda. • the section called “Pemantauan dengan CloudWatch”	15 November 2022
Rilis awal	Rilis awal Panduan Pengguna EventBridge Penjadwal.	10 November 2022

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