aws

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 yang dikelola. Sangat scalable, EventBridge Scheduler memungkinkan Anda untuk menjadwalkan jutaan tugas yang dapat memanggil lebih dari 270AWS 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 jangka waktu yang fleksibel untuk pengiriman, menentukan batas percobaan ulang, dan mengatur waktu retensi maksimum untuk pemicu yang gagal.

Topik

Fitur utama dari EventBridge Scheduler

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

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

- Jendela waktu yang 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 yang dijadwalkan secara 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 Penjadwal mencoba ulang tugas yang gagal dengan upaya tertunda untuk meningkatkan keandalan jadwal Anda dan memastikan target tersedia.

Mengakses EventBridge Scheduler

Anda dapat menggunakan EventBridge Scheduler melalui konsol EventBridge EventBridge Scheduler, SDK PenjadwalAWS CLI, atau dengan langsung menggunakan EventBridge Scheduler 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/signup.
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 Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Mmebuat 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 Identitas IAM (Direkomendasikan)	Gunakan kredensi jangka pendek untuk mengakses. AWS Ini sejalan dengan praktik terbaik keamanan. Untuk informasi tentang praktik terbaik, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.	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 admin IAM pertama Anda dan grup pengguna di Panduan Pengguna IAM.	Konfigurasikan akses terprogram dengan Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Gunakan kebijakan terkelola

Pada langkah sebelumnya, Anda mengatur pengguna IAM 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 prinsipal IAM Anda dengan cara yang sama seperti Anda melampirkan `AdministratorAccess` kebijakan pada langkah sebelumnya. Untuk informasi selengkapnya tentang mengelola akses ke EventBridge Scheduler menggunakan kebijakan IAM berbasis identitas, lihat. the section called “Menggunakan kebijakan berbasis identitas”

Mengatur peran eksekusi

Peran eksekusi adalah peran IAM yang diasumsikan oleh EventBridge Scheduler untuk berinteraksi dengan orang lain Layanan AWS 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 kebijakan JSON asumsi peran 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 izin JSON dan simpan secara lokal sebagai `.json` file.

   Amazon SQS – SendMessage

   Berikut ini memungkinkan EventBridge Scheduler untuk memanggil `sqs:SendMessage` tindakan pada semua antrian Amazon SQS 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 topik Amazon SNS 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 kebijakan ARN. Anda menggunakan ARN ini 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 ARN dari kebijakan yang Anda buat di langkah sebelumnya. 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 antrean Amazon SQS standar baru dengan. AWS CloudFormation

Untuk membuat antrean Amazon SQS baru

1. Salin AWS CloudFormation template JSON 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 antrean Amazon SQS 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 SDK 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 pengguna IAM dengan izin minimum yang diperlukan yang dijelaskan dalam. the 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 fileAWS 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 keAWS Management Console, lalu pilih tautan berikut untuk membuka bagian EventBridge Scheduler di EventBridge konsol: https://us-west-2.console.aws.amazon.com/scheduler/home?region=us-west-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. Misalnya, **`MyTestSchedule`**

   b. Untuk Deskripsi - opsional, masukkan deskripsi untuk jadwal Anda. Sebagai contoh, **`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 cron ekspresi. o gunakan ekspresi laju, pilih Jadwal berbasis tarif dan masukkan angka positif untuk Nilai, lalu pilih Unit dari opsi tarik-turun.

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.

> **Note**
>
> Fitur jendela waktu Fleksibel tidak tersedia dengan jadwal satu kali.

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 Selanjutnya.
7. Pada halaman Pilih target, lakukan hal berikut:

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

   b. Pada SendMessagebagian ini, untuk antrian SQS, pilih ARN antrian Amazon SQS yang ada `arn:aws:sqs:`*`us-west-2`*`:`*`123456789012`*`:`*`TestQueue`* seperti dari daftar drop-down. Untuk membuat antrean baru, pilih Buat antrean SQS baru untuk menavigasi ke konsol

Amazon SQS. Setelah selesai membuat antrian, kembali ke konsol EventBridge Scheduler dan segarkan drop-down. ARN antrian baru Anda 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 antrean 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 HAPUS jika Anda ingin jadwal dihapus secara otomatis. Untuk jadwal satu kali, ini terjadi setelah jadwal memanggil target satu kali. 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 maksimumnya adalah 24 jam.

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

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

f. Untuk antrian Dead-letter (DLQ), pilih dari opsi berikut:

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

g. Di bagian Enkripsi, pilih Sesuaikan pengaturan enkripsi (lanjutan) untuk menggunakan kunci KMS yang dikelola pelanggan untuk mengenkripsi input target Anda. Jika Anda memilih opsi ini, masukkan ARN kunci KMS yang ada atau pilih Buat AWS tombol KMS untuk menavigasi ke konsol. AWS KMS 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 konsol IAM 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 Selanjutnya.
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 target Amazon SQS, buka konsol Amazon SQS 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 untuk peran eksekusi yang ingin Anda kaitkan dengan jadwal.
- Arn — Masukkan ARN untuk target. Dalam hal ini, targetnya adalah antrian Amazon SQS.
- Input — Masukkan pesan yang EventBridge Scheduler kirimkan ke antrean 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 SDK EventBridge Scheduler

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

Example SDK Python

```
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 SDK Java

```
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 konsolAWS CLI, atau SDK EventBridge Penjadwal, lihat. Mengelola jadwal
- Untuk informasi selengkapnya tentang cara mengonfigurasi target template dan mempelajari cara menggunakan parameter target universal, lihatMengelola target.
- Untuk informasi selengkapnya tentang tipe data EventBridge Scheduler dan operasi API, lihat Referensi API EventBridge Scheduler.

Jenis jadwal pada 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 API target antara `1:00:00` dan`1:00:59`.

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 menyediakan jadwal berdasarkan tarif, jadwal Anda mulai memanggil target dengan segera. `StartDate`

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

Sintaksis

```
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 antrian Amazon SQS, 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. PST pada 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.

Sintaksis

```
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-DES	, - * /
D ay-of-week	1-7 atau MGG-SBT	, - * ? L #
Tahun	1970-2199	, - * /

Wildcard

- Wildcard , (koma) mencakup nilai tambahan. Di bidang Bulan, JAN, FEB, MAR mencakup 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 10:15 UTC+0 pada hari Jumat terakhir setiap bulan selama tahun 2022 hingga 2023, dan mengirimkan pesan ke antrian Amazon SQS, menggunakan jenis target template. `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.

Sintaksis

```
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 1 siang UTC-8 pada 20 November 2022, dan mengirimkan pesan ke antrian Amazon SQS, 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).

DenganAWS 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 target Amazon SQS template di America/New_York setiap hari pada `SendMessage` 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 daylight saving time 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

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 operasi API yang dipanggil EventBridge Scheduler atas nama Anda setiap kali jadwal Anda berjalan. EventBridge Scheduler mendukung dua jenis target: target template memanggil operasi API umum di seluruh grup layanan inti, 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 antrian 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. DLQ adalah EventBridge Queue Scheduler Amazon SQS standar yang digunakan untuk mengirimkan kejadian yang gagal, setelah kebijakan coba lagi habis. Anda dapat menggunakan DLQ untuk memecahkan masalah dengan jadwal Anda atau target hilirnya. Untuk informasi lebih lanjut tentang, lihatthe section called “Mengonfigurasi antrean surat mati”.

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

Topik

- Mengubah status jadwal
- Mengkonfigurasi jendela waktu yang fleksibel
- Mengonfigurasi antrean surat mati untuk jadwal
- Menghapus jadwal
- Apa selanjutnya?

Mengubah status jadwal

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 SDK Python

```
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 yang fleksibel

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 iniAWS CLI, dan contoh SDK 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 SDK Python

```
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)
```

Mengonfigurasi antrean surat mati untuk jadwal

Amazon EventBridge Scheduler mendukung antrian dead-letter (DLQ) menggunakan Amazon Simple Queue Service. Ketika jadwal gagal memanggil targetnya, EventBridge Scheduler mengirimkan payload JSON yang berisi detail pemanggilan dan respons apa pun yang diterima dari target ke antrean standar Amazon SQS yang Anda tentukan.

Topik berikut mengacu pada JSON ini sebagai peristiwa surat mati. Peristiwa dead-letter memungkinkan Anda memecahkan masalah dengan jadwal atau target Anda. Jika Anda mengonfigurasi kebijakan coba ulang untuk jadwal Anda, EventBridge Penjadwal mengirimkan peristiwa surat mati yang melelahkan jumlah percobaan ulang maksimum yang Anda tetapkan.

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

Topik

Buat antrean Amazon SQS

Sebelum mengonfigurasi DLQ untuk jadwal, Anda harus membuat antrean Amazon SQS standar. Untuk petunjuk cara membuat antrean menggunakan konsol Amazon SQS, lihat Membuat antrean Amazon SQS di Panduan Developer Amazon Simple Queue Service.

> **Note**
>
> EventBridge Penjadwal tidak mendukung penggunaan antrean FIFO sebagai DLQ jadwal Anda.

GunakanAWS CLI perintah berikut untuk membuat antrian standar.

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

Jika Anda berhasil, Anda akan melihat output.QueueURL

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

Setelah Anda membuat antrian, perhatikan antrian ARN. Anda akan memerlukan ARN saat Anda menentukan DLQ untuk EventBridge jadwal Penjadwal Anda. Anda dapat menemukan ARN antrean Anda di konsol Amazon SQS, atau dengan menggunakan `get-queue-attributes`AWS CLIperintah.

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

Jika Anda berhasil, Anda akan melihat antrean ARN di output.

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

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

Menyiapkan izin peran eksekusi

Agar EventBridge Scheduler mengirimkan peristiwa dead-letter ke Amazon SQS, 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 yang dilampirkan jika Anda menggunakan EventBridge Penjadwal untuk memanggil target API Amazon SQS.

Di bagian selanjutnya, Anda akan menggunakan konsol EventBridge Penjadwal dan menentukan DLQ untuk jadwal Anda.

Tentukan antrean surat mati

Untuk menentukan DLQ, gunakan konsol EventBridge Scheduler atauAWS CLI untuk memperbarui jadwal yang ada, atau membuat yang baru.

Console

Untuk menentukan DLQ menggunakan konsol

1. Masuk keAWS Management Console, lalu pilih link berikut untuk membuka bagian EventBridge Scheduler dari EventBridge conosle: https://console.aws.amazon.com/scheduler/home

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

AWS CLI

Untuk memperbarui jadwal yang ada menggunakanAWS CLI

- Gunakan `update-schedule`perintah untuk memperbarui jadwal Anda. Tentukan antrean Amazon SQS yang Anda buat sebelumnya sebagai DLQ. Tentukan ARN peran IAM yang Anda lampirkan izin Amazon SQS 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 menggunakanAWS 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 selanjutnya, Anda akan menggunakanAWS CLI untuk menerima peristiwa surat mati dari DLQ.

Mengambil surat mati

Gunakan `receive-message`perintah, seperti yang ditunjukkan dalam berikut ini, untuk mengambil peristiwa mati-huruf 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 Anda berhasil, Anda akan melihat output yang serupa dengan yang berikut.

```
{
    "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 diterima EventBridge Scheduler dari API layanan target. Dalam contoh sebelumnya, kode kesalahan yang dikembalikan oleh Amazon SQS`AWS.SimpleQueueService.NonExistentQueue`. Jika jadwal gagal memanggil target karena masalah dengan EventBridge Scheduler, Anda akan melihat kode galat berikut sebagai gantinya:`AWS.Scheduler.InternalServerError`.
- **`ERROR_MESSAGE`**- Berisi pesan kesalahan yang diterima EventBridge Scheduler dari API layanan 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 Penjadwal, Anda akan melihat pesan galat berikut sebagai gantinya:`Unexpected error occurred while processing the request`.
- **`TARGET_ARN`**- ARN target yang dipanggil jadwal Anda, dalam format ARN layanan berikut:`arn:aws:scheduler:::aws-sdk:service:apiAction`.
- **`EXHAUSTED_RETRY_CONDITION`**— Menunjukkan mengapa acara dikirim ke DLQ. Atribut ini akan hadir jika kesalahan dari API target 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 konfigurasi untuk jadwal`MaximumEventAgeInSeconds`, atau, jika acara lebih tua dari usia maksimum Anda dikonfigurasi pada jadwal dan masih gagal untuk memberikan.

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

Menghapus jadwal

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 HAPUS dari 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 file JSON. 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 SDK Python

```
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 data EventBridge Scheduler dan operasi API, lihat Referensi API EventBridge Scheduler.

Mengelola grup jadwal

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 diAkun 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 pengenal pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam tag. Tag dapat diakses oleh banyak layanan AWS, termasuk penagihan. Tag tidak dimaksudkan untuk digunakan dalam data sensitif atau privat.

Grup jadwal memiliki dua kemungkinan status: AKTIF dan MENGHAPUS.

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. dan Akhirnya, Anda akan menghapus grup.

Topik

- Membuat grup jadwal
- Menghapus grup jadwal
- Sumber daya terkait

Membuat grup jadwal

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. Sebagai contoh, **`TestGroup`**.
5. Di bagian Tag, lakukan hal berikut:
   a. Pilih Add new tag (Tambahkan tanda 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 menyelesaikannya, 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, lihatMulai.
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 tipe target Amazon SQS template untuk menjalankan `SendMessage` operasi. 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

Berikut ini, Anda dapat mempelajari cara menghapus grup jadwal menggunakan AWS Management Console danAWS 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 iniWilayah AWS, cari grup yang ingin Anda hapus. Jika Anda tidak melihat grup yang Anda cari, pilih yang lainWilayah AWS.

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

4. Pilih kotak centang untuk grup yang ingin Anda hapus.
5. Pilih Delete (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 Referensi API EventBridge Scheduler.
- DeleteScheduleGroupoperasi di Referensi API EventBridge Scheduler.

Mengelola target

Topik berikut menjelaskan cara menggunakan templated, dan target universal dengan EventBridge Scheduler, dan menyediakan daftarAWS layanan yang didukung yang dapat Anda konfigurasikan menggunakan parameter target universal EventBridge Scheduler.

Target templated adalah serangkaian operasi API umum di seluruh grupAWS layanan inti seperti Amazon SQS, Lambda, dan Step Functions. Misalnya, Anda dapat menargetkan operasi API Invoke Lambda dengan menyediakan fungsi ARN, atau `SendMessage`operasi Amazon SQS dengan ARN antrean target.

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

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

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

Topik

Menggunakan target template

Target template adalah serangkaian operasi API umum di seluruh grup AWS layanan inti, seperti Amazon SQS, Lambda, dan Step Functions. Misalnya, Anda dapat menargetkan `Invoke`operasi Lambda dengan menyediakan fungsi ARN, atau operasi Amazon SQS menggunakan ARN `SendMessage`antrian. Untuk mengonfigurasi target template, Anda juga harus memberikan izin ke peran eksekusi jadwal untuk melakukan operasi API yang ditargetkan.

Untuk mengonfigurasi target template secara terprogram menggunakan AWS CLI atau salah satu SDK EventBridge Scheduler, Anda perlu menentukan ARN peran eksekusi, ARN untuk sumber daya target, 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. Saat Anda menentukan ARN untuk sumber daya target template, EventBridge Scheduler secara otomatis mengasumsikan bahwa Anda ingin memanggil operasi API yang didukung untuk layanan tersebut. Jika Anda ingin EventBridge Scheduler menargetkan operasi API 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 Referensi API 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 izin IAM 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 SDK Python

```
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 SDK Java

```
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 SDK Python

```
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 SDK Java

```
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 SDK Python

```
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 SDK Java

```
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

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

Untuk mengonfigurasi target universal untuk jadwal Anda menggunakanAWS CLI, atau salah satu SDK EventBridge Penjadwal, 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 operasi API yang ingin ditargetkan oleh jadwal Anda.
- ARN — ARN layanan lengkap, termasuk operasi API yang ingin Anda targetkan, dalam format berikut:. `arn:aws:scheduler:::aws-sdk:`*`service`*`:`*`apiAction`*

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

Tindakan yang tidak didukung

EventBridge Scheduler tidak mendukung tindakan API 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
isAuthorizedWithToken
invokeModel
```

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

Contoh menggunakan target universal

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

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

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

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>**— ARN dari jadwal.
- **<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 diaktifkan setiap lima menit, dan memanggil operasi Amazon SendMessage SQS sebagai target universal. Badan pesan mencakup nilai untukschedule-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 SDK Python

```
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 data EventBridge Scheduler dan operasi API, lihat Referensi APIEventBridge Penjadwal.

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 Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber Penjadwal. EventBridge IAM adalah Layanan AWS 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, maka administrator Anda memberi Anda kredensyal 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. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat

menggunakan IAM dengan EventBridge Scheduler, lihat. Bagaimana EventBridge Scheduler bekerja dengan IAM

Administrator IAM — Jika Anda seorang administrator IAM, 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 di IAM, lihat. Menggunakan kebijakan berbasis identitas

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

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

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

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

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

Akun AWS pengguna root

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

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensyal 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 Layanan AWS 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 Identitas IAM, 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 Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin ke grup tersebut untuk mengelola sumber daya IAM.

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

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

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

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

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

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

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

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

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

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

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat 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 yang dikelola atau kebijakan inline, lihat Memilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

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

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

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL) dalam Panduan Developer Amazon Simple Storage Service.

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.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran 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 batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.

- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON 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 di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai 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 Pengguna IAM.

Bagaimana EventBridge Scheduler bekerja dengan IAM

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

Fitur IAM yang dapat Anda gunakan dengan Amazon Scheduler EventBridge

Fitur IAM	EventBridge Dukungan penjadwal
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya

Fitur IAM	EventBridge Dukungan penjadwal
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACL	Tidak
ABAC (tanda 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 Scheduler dan AWS layanan lainnya dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Scheduler EventBridge

Mendukung kebijakan berbasis identitas	Ya

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

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. 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 kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Scheduler EventBridge

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

Kebijakan berbasis sumber daya dalam Scheduler EventBridge

Mendukung kebijakan berbasis sumber daya	Tidak

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

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM 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, administrator IAM 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 di IAM di Panduan Pengguna IAM.

Tindakan kebijakan untuk EventBridge Scheduler

Mendukung tindakan kebijakan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. 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 kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, 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 ARNnya, 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

Kunci kondisi kebijakan untuk EventBridge Scheduler

Mendukung kunci kondisi kebijakan khusus layanan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan 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. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tag dalam Panduan Pengguna IAM.

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 Pengguna IAM.

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

ACL di Scheduler EventBridge

Mendukung ACL	Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Scheduler EventBridge

Mendukung ABAC (tanda dalam kebijakan)	Parsial

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tag milik prinsipal cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat 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 selengkapnya tentang ABAC, lihat Apa itu ABAC? dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensyal sementara dengan Scheduler EventBridge

Mendukung penggunaan kredensial sementara	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensyal sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensyal sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

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 tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensyal sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Peralihan peran (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensyal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensyal 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 Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk Scheduler EventBridge

Mendukung sesi akses maju (FAS)	Ya

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

Peran layanan untuk EventBridge Scheduler

Mendukung peran layanan	Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **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. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan 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

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 di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

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

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh EventBridge Penjadwal, termasuk format ARN 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 yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA dalam Panduan Pengguna IAM.

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

EventBridge Izin penjadwal

Agar prinsipal IAM (pengguna, grup, atau peran) dapat membuat jadwal di EventBridge Penjadwal dan mengakses sumber EventBridge Penjadwal melalui konsol atau API, 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 API. `CreateSchedule` 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`
- Jadwal 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 kebijakan IAM 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 lebih lanjut, lihat AWS kebijakan terkelola dalam Panduan Pengguna

IAM. 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 kunci KMS tertentu yang akan digunakan untuk mengenkripsi data pada Scheduler. EventBridge 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 “AmazonEventBridgeS	EventBridge Scheduler menambahkan dukungan	10 November 2022

Perubahan	Deskripsi	Tanggal
chedulerReadOnlyAc cess ”— Kebijakan terkelola baru	untuk kebijakan terkelola baru yang memberi pengguna akses hanya-baca ke semua sumber daya, termasuk jadwal, dan grup jadwal.	
EventBridge Scheduler mulai melacak perubahan	EventBridge Scheduler mulai melacak perubahan untuk kebijakan yang AWS dikelola.	10 November 2022

Pencegahan Deputi Bingung

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 confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Kondisi berikut dicakup oleh kelompok jadwal individu: `arn:aws:scheduler:*:`*`123456789012`*`:schedule-group/`*`your-schedule-group`*

Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global

dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya: `arn:aws:scheduler:*:123456789012:schedule-group/*`.

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

> **Important**
>
> Jangan lingkup `aws:SourceArn` pernyataan ke jadwal tertentu atau awalan nama jadwal. ARN yang 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 dan IAM.

Topik

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 pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya fiktif *`my-example-widget`*, tetapi tidak memiliki izin fiktif `scheduler:`*`GetWidget`*.

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

Contoh kesalahan berikut terjadi ketika pengguna IAM 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 (ACL), 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 pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- 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 Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di 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 Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

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

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan EventBridge Scheduler atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Topik

- Enkripsi diam

- Enkripsi bergerak

Enkripsi diam

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 kunci KMS: Kunci milik AWS, dan kunci yang dikelola pelanggan.

> **Note**
>
> EventBridge Scheduler hanya mendukung penggunaan kunci KMS enkripsi simetris.

Kunci milik AWSadalah kunci KMS 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 kunci KMS 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
- Mengelola kunci KMS

- CloudTrail contoh acara

Artefak enkripsi

Tabel berikut menjelaskan berbagai jenis data yang EventBridge Scheduler mengenkripsi saat istirahat, dan jenis kunci KMS 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
Pengidentifikasi 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 pemanggilan satu kali, serta tanggal mulai jadwal, tanggal akhir, dan zona waktu.	Didukung	Tidak Support
Konfigurasi target	Nama Sumber Daya Amazon (ARN) target, dan detail konfigurasi terkait target lainnya.	Didukung	Tidak Support

Tipe data	Deskripsi	Kunci milik AWS	kunci yang dikelola pelanggan
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 kunci KMS” bagian berikut untuk mempelajari bagaimana Anda harus mengelola sumber daya IAM dan kebijakan utama agar dapat menggunakan kunci yang dikelola pelanggan dengan EventBridge Scheduler.

Mengelola kunci KMS

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 izin IAM 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 Scheduler API. 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 kunci dan izin IAM yang diperlukan untuk menggunakan kunci terkelola pelanggan di EventBridge Scheduler.

Topik

- Tambahkan izin IAM
- Kelola kebijakan utama

Tambahkan izin IAM

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

Izin berbasis identitas untuk kunci yang dikelola pelanggan

Anda harus menambahkan AWS KMS tindakan berikut ke kebijakan izin yang terkait dengan prinsipal (pengguna, grup, atau peran) yang memanggil EventBridge Scheduler 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 KMS enkripsi simetris.
- **`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.

Kelola 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 peristiwa panggilan API. Ini termasuk panggilan API setiap kali EventBridge Scheduler menggunakan kunci terkelola 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 bergerak

EventBridge Scheduler mengenkripsi data Anda dalam perjalanan saat melakukan perjalanan jaringan. Transport Layer Security (TLS) mengenkripsi data Anda saat Anda memanggil operasi EventBridge Scheduler API apa pun, serta saat EventBridge Scheduler memanggil API target apa pun 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 versi TLS yang berbeda saat menggunakan EventBridge Scheduler.

Menggunakan EventBridge Scheduler API — Saat Anda menjalankan operasi API, seperti`CreateSchedule`, EventBridge Scheduler mengenkripsi seluruh permintaan HTTP, termasuk isi permintaan dan header. EventBridge Scheduler juga mengenkripsi seluruh objek respons yang Anda terima dari API kami.

Menggunakan API target — Saat EventBridge Scheduler memanggil jadwal Anda, Scheduler memanggil API target 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 Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS 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 Layanan AWS 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 Keamanan dan Kepatuhan HIPAA di Amazon Web Services — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

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

- 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 Layanan AWS 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 Layanan AWS 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 Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit ManagerIni Layanan AWS 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 panggilan API yang AWS dipublikasikan untuk mengakses EventBridge Scheduler melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

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

Pemantauan dan metrik untuk Amazon EventBridge Scheduler

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa Amazon EventBridge Scheduler danAWS solusi Anda lainnya. AWSmenyediakan alat pemantauan berikut untuk mengawasi EventBridge Scheduler, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan:

- Amazon CloudWatch memantauAWS sumber daya Anda dan aplikasi yang Anda jalankanAWS di secara waktu nyata. 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 CloudTrail merekam panggilan API dan peristiwa terkait yang dilakukan oleh atau atas nama akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang memanggil AWS, alamat IP sumber yang melakukan panggilan, dan kapan panggilan tersebut terjadi. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.

Topik

Memantau Amazon EventBridge Scheduler 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 surat mati terkait (DLQ). Jika Anda mengonfigurasi DLQ untuk jadwal Anda, EventBridge Scheduler akan menerbitkan metrik tambahan saat jadwal Anda habis kebijakan coba ulangnya.

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

Sebuah 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, InvocationAttemptHitung atau InvocationDroppedHitung, 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 jadwal EventBridge 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 EventBrid ge Scheduler mencoba memanggil jadwal Anda, dan untuk melihat kapan pemanggilan mendekati kuota akun Anda.
`AWS/Scheduler`	`TargetErr orCount`	Hitung	Dipancarkan saat target mengembal ikan pengecualian setelah EventBridge Scheduler memanggil API target. Gunakan ini untuk memeriksa kapan pengiriman ke target gagal.
`AWS/Scheduler`	`TargetErr orThrottl edCount`	Hitung	Dipancarkan saat pemanggilan target gagal karena pelambatan API oleh target. Gunakan ini untuk mendiagnosis kegagalan pengirima n ketika alasan dasarnya adalah panggilan pembatasa n API target yang dilakukan oleh Scheduler EventBrid ge

Namespace	Metrik	Unit	Deskripsi
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 EventBridge Scheduler Anda. Untuk informasi selengkapnya tentang kuota layanan, lihatKuota.
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 API EventBridge Scheduler.

Metrik untuk jadwal dengan DLQ

Namespace	Metrik	Unit	Deskripsi
AWS/Scheduler	InvocationsSentToDeadLetterCount	Hitung	Dipancarkan untuk setiap pengiriman yang berhasil ke DLQ jadwal. Gunakan ini untuk menentukan kapan acara dikirim ke DLQ, lalu periksa acara yang dikirimkan ke DLQ jadwal untuk detail tambahan yang membantu Anda menentukan penyebab kegagalan.
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount	Hitung	Dipancarkan saat EventBridge Scheduler tidak dapat mengirimkan acara ke DLQ.
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount_<error_code>	Hitung	

Namespace	Metrik	Unit	Deskripsi
			Gunakan dua metrik ini untuk menentukan alasan mengapa EventBridge Scheduler tidak dapat mengirim acara ke DLQ, dan memodifikasi konfigurasi DLQ Anda untuk menyelesaikan masalah. Berikut ini adalah contoh `InvocationsFailedToBeSentToDeadLetterCount_<error_code>` metrik saat antrian Amazon SQS yang Anda tentukan sebagai DLQ tidak ada:

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

Penjadwal EventBridge metrik penggunaan

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 selengkapnya 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`. Dimensi `Resource` menentukan nama operasi API yang sedang ditelusuri.

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

- “Layanan”: “Penjadwal”
- “Jenis”: “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. Penjadwal EventBridge Metrik penggunaan API 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 operasi API. Nilai-nilai yang valid meliputi: • `CreateSchedule` • `CreateScheduleGroup` • `DeleteSchedule` • `DeleteScheduleGroup` • `GetSchedule` • `GetScheduleGroup` • `ListScheduleGroups` • `ListSchedulesCallCount`

Dimensi	Deskripsi		
	• `ListTagsForResource` • `TagResource` • `UntagResource` • `UpdateSchedule`		

Menggunakan log panggilan API Amazon EventBridge SchedulerAWS CloudTrail

Amazon EventBridge Scheduler terintegrasi denganAWS CloudTrail, layanan yang menyediakan layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWS layanan di EventBridge Scheduler. CloudTrail merekam semua panggilan API untuk EventBridge Scheduler sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari konsol EventBridge Scheduler dan panggilan kode ke operasi API EventBridge Scheduler. Jika membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan dari CloudTrail kejadian ke bucket Amazon S3, termasuk peristiwa untuk EventBridge Scheduler. Jika Anda tidak dapat mengonfigurasi jejak, Anda masih dapat melihat kejadian terbaru di konsol di Riwayat peristiwa peristiwa terbaru di CloudTrail konsol di Riwayat peristiwa peristiwa. Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke EventBridge Scheduler, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjut CloudTrail, lihat PanduanAWS CloudTrail Pengguna.

EventBridge Informasi penjadwal di CloudTrail

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

Untuk catatan berkelanjutan tentang peristiwa di AndaAkun AWS, termasuk peristiwa untuk EventBridge Scheduler, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim file log ke bucket Amazon S3 Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS

dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasiAWS layanan lainnya untuk dianalisis lebih lanjut dan bertindak berdasarkan data kejadian yang dikumpulkan di CloudTrail log. Untuk informasi selengkapnya, lihat yang berikut:

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

Semua tindakan EventBridge Penjadwal dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API Amazon EventBridge Scheduler. 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 hal berikut:

- Bahwa permintaan dibuat dengan kredensial pengguna root atau pengguna AWS Identity and Access Management (IAM).
- Bahwa permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

EventBridge Memahami entri file log

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau beberapa entri log Sebuah peristiwa mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut 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 sebaliknya, setiap kuota unik untuk suatu Wilayah. Anda dapat meminta kenaikan untuk beberapa kuota, dan 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 Kuota Layanan, gunakan formulir penambahan batas.

> **Note**
>
> Kuota `CreateScheduleUpdateSchedule`,`GetSchedule`,, dan `DeleteSchedule` transaksi per detik (TPS) untuk EventBridge Scheduler dapat disesuaikan hingga ribuan TPS. Kuota throttle pemanggilan dapat disesuaikan hingga puluhan ribu TPS.

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

Nama	Default	Dapat disesuaikan	Deskripsi
CreateSchedule tingkat permintaan	Setiap Wilayah yang didukung: 50	Ya	CreateSchedule Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
CreateScheduleGroup tingkat permintaan	Setiap Wilayah yang didukung: 10	Ya	CreateScheduleGroup Permintaan maksimum per detik. Ketika Anda

Nama	Default	Dapat disesuaikan	Deskripsi
			mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
DeleteSchedule tingkat permintaan	Setiap Wilayah yang didukung: 50	Ya	DeleteSchedule Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.
DeleteScheduleGroup tingkat permintaan	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	Setiap Wilayah yang didukung: 50	Ya	GetSchedule Permintaan maksimum per detik. Ketika Anda mencapai kuota ini, EventBridge Scheduler menolak permintaan untuk operasi ini selama sisa interval.

Nama	Default	Dapat disesu an	Deskripsi
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.
Batas throttle pemanggilan dalam transaksi per detik	Setiap Wilayah yang didukung: 500	Ya	Doa adalah payload jadwal yang dikirimkan ke target yang ditentuka n. 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.

Nama	Default	Dapat disesuaikan	Deskripsi
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.
Jumlah jadwal	Setiap Wilayah yang didukung: 1.000.000	Ya	Jumlah maksimum jadwal per wilayah. Kuota ini termasuk jadwal satu kali yang telah selesai berjalan. Sebaiknya hapus jadwal satu kali Anda setelah selesai berjalan dan memanggil target.
TagResource tingkat permintaan	Setiap Wilayah yang didukung: 1	Ya	Menetapkan satu atau beberapa tag (pasangan kunci-nilai) ke sumber Scheduler yang ditentukan.

Nama	Default	Dapat disesu an	Deskripsi
UntagResource tingkat permintaan	Setiap Wilayah yang didukung: 1	Ya	Menghapus satu atau beberapa tag dari sumber Scheduler yang ditentuka n.
UpdateSchedule tingkat permintaan	Setiap Wilayah yang didukung: 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

Riwayat dokumen untuk EventBridge Panduan Pengguna Penjadwal

Tabel berikut ini menjelaskan performa untuk EventBridge Penjadwal.

Perubahan	Deskripsi	Tanggal
Perubahan peran eksekusi dan pencegahan wakil yang membingungkan	Pembaruan 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”	September 7, 2023
Penghapusan otomatis jadwal 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”	Agustus 2, 2023
Topik terbaru tentang penggunaan target universal	Memperbarui daftar layanan yang didukung yang EventBridge Scheduler dapat menargetkan dan mengintegrasikan dengan. Pembaruan ini juga	Maret 17, 2023

	mencakup daftar yang tidak didukungGETOperasi API, 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 Penjadwal. Gunakan chapter ini untuk mempelajari cara membuat grup, menambahkan jadwal ke grup, menerapkan tag untuk lebih mudah mengelola dan monirot Anda EventBridge Sumber daya penjadwal, dan akhirnya menghapus grup. • Mengelola grup jadwal	Maret 17, 2023

Topik baru tentang waktu musim panas dan zona waktu	Ditambahkan 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”	November 17, 2022
Topik baru tentang metrik	Menambahkan topik baru yang menjelaskan metrik yang EventBridge Scheduler menerbitkan ke CloudWatch. Anda dapat menggunakan metrik ini untuk memantau performa Anda. • the section called “Pemantauan CloudWatch dengan”	November 15, 2022
Rilis awal	Rilis awal EventBridge Panduan Pengguna Penjadwal.	November 10, 2022

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