Amazon CloudWatch Peristiwa

Panduan Pengguna

Amazon CloudWatch Peristiwa: Panduan Pengguna

Table of Contents

Apa Itu Amazon CloudWatch Events? 1
- Konsep 1
- Layanan AWS Terkait 2

Mengatur 4
- Daftar ke Amazon Web Services (AWS) 4
- Masuk ke Konsol Amazon CloudWatch 4
- Kredensial Akun 4
- Siapkan Antarmuka Baris Perintah 5
- Titik Akhir Regional 5

Memulai 6
- Membuat Aturan yang Memicu Peristiwa 7
- Buat Aturan yang Memicu Panggilan API AWS melalui CloudTrail 8
- Membuat Aturan yang Memicu Jadwal 9
- Menonaktifkan atau Menghapus Aturan 10

Tutorial 11
- Tutorial: Relai Systems Manager Run Command 11
- Tutorial: Catat Status Instans EC2 12
  - Langkah 1: BuatAWS LambdaFungsi 13
  - Langkah 2: Buat aturan 13
  - Langkah 3: Uji Aturan 14
- Tutorial: Log Status Grup Auto Scaling 14
  - Langkah 1: BuatAWS LambdaFungsi 14
  - Langkah 2: Buat aturan 15
  - Langkah 3: Uji Aturan 16
- Tutorial: Log Operasi Tingkat Objek S3 16
  - Langkah 1: Konfigurasi AndaAWS CloudTrailTrail 16
  - Langkah 2: BuatAWS LambdaFungsi 17
  - Langkah 3: Buat aturan 18
  - Langkah 4: Uji Aturan 18
- Tutorial: Gunakan Transformator Input untuk Menyesuaikan Apa yang Diteruskan ke Target Peristiwa 19
  - Buat aturan 19
- Tutorial: LogAWSPanggilan API 20
  - Prasyarat 20
  - Langkah 1: BuatAWS LambdaFungsi 21
  - Langkah 2: Buat aturan 21
  - Langkah 3: Uji Aturan 22
- Tutorial: Jadwalkan Snapshot EBS Otomatis 22
  - Langkah 1: Buat aturan 22
  - Langkah 2: Uji Aturan 23
- Tutorial: Jadwalkan Fungsi Lambda 23
  - Langkah 1: BuatAWS LambdaFungsi 24
  - Langkah 2: Buat aturan 24
  - Langkah 3: Verifikasi Aturan 26
- Tutorial: Mengatur Systems Manager Automation sebagai Target 26
- Tutorial: Relai Peristiwa ke Aliran Kinesis 27
  - Prasyarat 27
  - Langkah 1: Membuat Amazon Kinesis Stream 27
  - Langkah 2: Buat aturan 27
  - Langkah 3: Uji Aturan 28
  - Langkah 4: Verifikasi bahwa Peristiwa Direlai 28
- Tutorial: Jalankan Tugas Amazon ECS ketika File Diunggah ke Bucket Amazon S3 29
- Tutorial: Jadwalkan Pembangunan Otomatis Menggunakan CodeBuild 30
- Tutorial: Mencatat Perubahan Status Instans Amazon EC2 31

Ekspresi Jadwal untuk Aturan 33

Ekspresi Cron 33
Ekspresi Rate 36
Pola Peristiwa 37
Pola Peristiwa 38
Pencocokan Nilai Null dan String Kosong dalam Pola Peristiwa 39
Array dalam Pola Peristiwa 40
Peristiwa yang Layanan Didukung 42
Peristiwa Amazon Augmented AI 43
Peristiwa Application Auto Scaling 43
Peristiwa AWS Batch 43
AmazonCloudWatchPeristiwa terjadwal 43
Peristiwa Amazon Chime 44
Peristiwa dariCloudWatch 44
CodeBuildPeristiwa 44
CodeCommitPeristiwa 44
Peristiwa AWS CodeDeploy 44
CodePipelinePeristiwa 45
Peristiwa AWS Config 46
Peristiwa Amazon EBS 46
Peristiwa Amazon EC2 Auto Scaling 47
Peristiwa Rekomendasi Penyeimbangan Ulang Instans Amazon EC2 47
Peristiwa Interupsi Instans Spot Amazon EC2 47
Peristiwa Perubahan Status Amazon EC2 47
Peristiwa Amazon ECR 47
Peristiwa Amazon ECS 48
AWS ElementalMediaConvertPeristiwa 48
AWS ElementalMediaPackagePeristiwa 48
AWS ElementalMediaStorePeristiwa 48
Peristiwa Amazon EMR 48
AmazonGameLiftPeristiwa 50
Peristiwa AWS Glue 57
Peristiwa AWS Ground Station 62
Amazon GuardDuty Events 62
Peristiwa AWS Health 62
Peristiwa AWS KMS 64
Peristiwa Amazon Macie 65
Peristiwa Masuk AWS Management Console 65
Peristiwa Tumpukan AWS OpsWorks 66
SageMakerPeristiwa 68
Peristiwa AWS Security Hub 68
Peristiwa AWS Server Migration Service 68
Peristiwa AWS Systems Manager 69
Peristiwa Otomatisasi AWS Systems Manager 70
Peristiwa Kalender Perubahan AWS Systems Manager 70
Peristiwa Kepatuhan AWS Systems Manager 71
Peristiwa Windows Maintenance AWS Systems Manager 73
Peristiwa Menyimpan Parameter AWS Systems Manager 75
Peristiwa Run Command AWS Systems Manager 76
Peristiwa State Manager AWS Systems Manager 77
Peristiwa AWS Step Functions 78
Tandai Perubahan Peristiwa di Sumber Daya AWS 78
Peristiwa AWS Trusted Advisor 78
WorkSpacesPeristiwa 80
Peristiwa yang Disampaikan ViaCloudTrail 80
Mengirim dan Menerima Peristiwa Antara Akun AWS 82
Mengaktifkan Akun AWS untuk Menerima Peristiwa dari Akun AWS Lainnya 83
Mengirim Peristiwa ke Akun AWS Lain 84

Menulis Aturan yang Cocok dengan Peristiwa dari Akun AWS Lain 86
Memigrasi Hubungan Pengirim-Penerima untuk Menggunakan AWS Organizations 87
Menambahkan Peristiwa dengan PutEvents 89
Menangani Kegagalan Saat Menggunakan PutEvents 90
Mengirim Peristiwa Menggunakan AWS CLI 91
Menghitung Ukuran Entri Peristiwa PutEvents 91
Menggunakan CloudWatch Events dengan VPC Endpoint Antarmuka 93
Ketersediaan 93
Membuat VPC Endpoint untuk CloudWatch Events 94
Mengendalikan Akses ke VPC Endpoint CloudWatch Events 94
Pemantauan Penggunaan dengan Metrik CloudWatch 96
Metrik CloudWatch Events 96
Dimensi untuk Metrik CloudWatch Events 96
Aturan Terkelola 98
Bekerja denganAWSSDK 99
Contoh kode 100
Tindakan 100
Menambahkan target fungsi Lambda 100
Membuat aturan terjadwal 103
Kirim acara 105
Keamanan 108
MenandaiCloudWatchAcara Sumber Daya 109
Sumber Daya yang DidukungCloudWatchPeristiwa 109
Mengelola Tag 109
Kesepakatan Penamaan dan Penggunaan Tag 110
Mencatat Panggilan API 111
Informasi CloudWatch Events di CloudTrail 111
Contoh: CloudWatch Catatan File Log 112
Service Quotas 114
Pemecahan Masalah 115
Aturan saya dipicu tapi fungsi Lambda saya tidak dipanggil 115
Saya baru saja membuat atau memodifikasi aturan, tetapi tidak cocok dengan peristiwa pengujian 116
Aturan saya tidak memicu sendiri pada waktu yang ditentukan dalam ScheduleExpression 117
Aturan saya tidak memicu pada waktu yang saya harapkan 117
Aturan saya cocok dengan panggilan API IAM tapi aturan saya tidak dipicu 117
Aturan saya tidak bekerja karena IAM role yang terkait dengan aturan diabaikan ketika aturan dipicu 118
Saya membuat aturan dengan EventPattern yang seharusnya cocok dengan sumber daya, tapi saya tidak melihat peristiwa yang cocok dengan aturan 118
Pengiriman peristiwa saya ke target tertunda 118
Beberapa kejadian tidak pernah dikirimkan ke target saya 118
Aturan saya dipicu lebih dari sekali dalam menanggapi satu peristiwa. Jaminan apa yang ditawarkan CloudWatch Events untuk memicu aturan atau menyampaikan peristiwa ke target? 119
Mencegah Loop Tak Terbatas 119
Kejadian saya tidak dikirim ke antrean Amazon SQS target 119
Aturan saya dipicu, tapi saya tidak melihat pesan yang diterbitkan ke topik Amazon SNS saya 120
Topik Amazon SNS saya masih memiliki izin untuk CloudWatch Events bahkan setelah saya menghapus aturan yang terkait dengan topik Amazon SNS 121
Kunci syarat IAM mana yang dapat saya gunakan dengan CloudWatch Events? 121
Bagaimana saya bisa tahu saat aturan CloudWatch Events rusak? 121
Riwayat Dokumen 123
Daftar istilah AWS 126
..... cxxvii

Apa Itu Amazon CloudWatch Events?

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Amazon CloudWatch Events menyediakan aliran sistem nyaris secara langsung yang menguraikan perubahan dalam sumber daya Amazon Web Services (AWS). Dengan menggunakan aturan sederhana yang dapat Anda siapkan dengan cepat, Anda dapat mencocokkan acara dan merutekannya ke satu atau beberapa fungsi atau pengaliran target. CloudWatch Events akan menyadari perubahan operasional yang terjadi. CloudWatch Events merespons perubahan operasional ini dan mengambil tindakan korektif yang diperlukan, dengan mengirim pesan untuk merespons lingkungan, mengaktifkan fungsi, membuat perubahan, dan merekam informasi status.

Anda juga dapat menggunakan CloudWatch Events untuk menjadwalkan tindakan otomatis yang terpicu sendiri pada waktu tertentu menggunakan ekspresi cron atau laju. Untuk informasi selengkapnya, lihat Ekspresi Jadwal untuk Aturan (p. 33).

Anda dapat mengonfigurasi layanan AWS berikut sebagai target untuk CloudWatch Events:

- Instans Amazon EC2
- AWS Lambda fungsi
- Aliran di Amazon Kinesis Data Streams
- Aliran pengiriman di Amazon Kinesis Data Firehose
- Grup log di Amazon CloudWatch Logs
- Tugas Amazon ECS
- Run Command Systems Manager
- Otomatisasi Systems Manager
- Pekerjaan AWS Batch
- Mesin status Step Functions
- Alur dalam CodePipeline
- Proyek CodeBuild
- Templat penilaian Amazon Inspector
- Topik Amazon SNS
- Antrean Amazon SQS
- Target bawaan: `EC2 CreateSnapshot API call`, `EC2 RebootInstances API call`, `EC2 StopInstances API call`, dan `EC2 TerminateInstances API call`.
- Bus peristiwa default dari akun AWS lain

Konsep

Sebelum Anda mulai menggunakan CloudWatch Events, Anda harus memahami konsep berikut:

- Peristiwa – Sebuah peristiwa menunjukkan perubahan di lingkungan AWS Anda. Sumber daya AWS dapat menghasilkan peristiwa ketika statusnya berubah. Misalnya, Amazon EC2 menghasilkan peristiwa ketika status instans EC2 berubah dari tertunda menjadi berjalan, dan Amazon EC2 Auto Scaling menghasilkan peristiwa ketika meluncurkan atau mengakhiri instans. AWS CloudTrail memublikasikan peristiwa ketika Anda membuat panggilan API. Anda dapat membuat peristiwa kustom tingkat aplikasi dan memublikasikannya ke CloudWatch Events. Anda juga dapat mengatur peristiwa terjadwal yang dihasilkan secara berkala. Untuk daftar layanan yang menghasilkan peristiwa, termasuk contoh peristiwa dari setiap layanan, lihat CloudWatchContoh Peristiwa dari Layanan yang Didukung (p. 42).
- Aturan – Aturan mencocokkan peristiwa yang masuk dan merutekannya ke target untuk pemrosesan. Satu aturan dapat merutekan ke beberapa target, yang semuanya diproses secara paralel. Aturan tidak diproses dalam urutan tertentu. Hal ini memungkinkan berbagai bagian dari organisasi untuk mencari dan memproses peristiwa yang sesuai bagi mereka. Aturan dapat menyesuaikan JSON yang dikirim ke target, dengan hanya meneruskan bagian-bagian tertentu atau menimpanya dengan konstanta.
- Target – Target memproses peristiwa. Target dapat berupa instans Amazon EC2, fungsi AWS Lambda, aliran Kinesis, tugas Amazon ECS, mesin status Step Functions, topik Amazon SNS, antrean Amazon SQS, dan target bawaan. Target menerima peristiwa dalam format JSON.

  Target aturan harus berada di Wilayah yang sama dengan aturan.

Layanan AWS Terkait

Layanan berikut digunakan bersama dengan CloudWatch Events:

- AWS CloudTrail memungkinkan Anda untuk memantau panggilan yang dilakukan ke API CloudWatch Events untuk akun Anda, termasuk panggilan yang dilakukan oleh AWS Management Console, AWS CLI, dan layanan lainnya. Ketika pencatatan CloudTrail diaktifkan, CloudWatch Events menulis berkas log ke bucket S3. Setiap berkas log berisi satu atau beberapa catatan, tergantung pada berapa banyak tindakan yang dilakukan untuk memenuhi permintaan. Untuk informasi selengkapnya, lihat Mencatat Panggilan API Amazon CloudWatch Events dengan AWS CloudTrail (p. 111).
- AWS CloudFormation memungkinkan Anda untuk memodelkan dan menyiapkan sumber daya AWS. Anda membuat templat yang menjelaskan sumber daya AWS yang Anda inginkan, dan AWS CloudFormation yang akan mengurus penyediaan dan konfigurasi sumber daya tersebut untuk Anda. Anda dapat menggunakan aturan CloudWatch Events di templat AWS CloudFormation Anda. Untuk informasi selengkapnya, lihat ::Aturan::Peristiwa::AWS dalam Panduan Pengguna AWS CloudFormation.
- AWS Config memungkinkan Anda untuk merekam perubahan konfigurasi pada sumber daya AWS. Ini mencakup bagaimana sumber daya terkait satu sama lain dan bagaimana konfigurasi sumber daya tersebut sebelumnya, sehingga Anda dapat melihat perubahan konfigurasi dan hubungan di antaranya dari waktu ke waktu. Anda juga dapat membuat aturan AWS Config untuk memeriksa apakah sumber daya Anda patuh atau tidak pada kebijakan organisasi Anda. Untuk informasi selengkapnya, lihat Panduan Developer AWS Config.
- AWS Identity and Access Management (IAM) membantu Anda mengontrol akses ke sumber daya AWS dengan aman untuk pengguna Anda. Gunakan IAM untuk mengontrol siapa saja yang dapat menggunakan sumber daya AWS Anda (autentikasi), sumber daya apa yang dapat mereka gunakan, dan bagaimana mereka dapat menggunakannya (otorisasi). Untuk informasi selengkapnya, lihatPanduan Pengguna IAM.
- Amazon Kinesis Data Streams memungkinkan pengambilan dan agregasi data yang cepat secara hampir terus-menerus. Jenis data yang digunakan meliputi data log infrastruktur IT, log aplikasi, media sosial, umpan data pasar, dan data clickstream web. Karena waktu respons untuk pengambilan dan pengolahan data secara waktu nyata, pemrosesannya biasanya ringan. Untuk informasi selengkapnya, lihat Panduan Developer Amazon Kinesis Data Streams.
- AWS Lambda memungkinkan Anda membangun aplikasi yang merespons informasi baru dengan cepat. Unggah kode aplikasi Anda sebagai fungsi Lambda dan Lambda akan menjalankan kode Anda pada infrastruktur komputasi ketersediaan tinggi. Lambda melakukan semua administrasi sumber daya komputasi, termasuk pemeliharaan server dan sistem operasi, penyediaan kapasitas dan penskalaan

otomatis, deployment patch kode dan keamanan, serta pemantauan dan pencatatan kode. Untuk informasi selengkapnya, lihat Panduan Developer AWS Lambda.

Menyiapkan Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Untuk menggunakan Amazon CloudWatch Events, Anda memerlukan akun AWS. Akun AWS Anda mengizinkan Anda menggunakan layanan (misalnya, Amazon EC2) untuk menghasilkan peristiwa yang dapat dilihat di konsol CloudWatch, antarmuka berbasis web. Selain itu, Anda dapat menginstal dan mengonfigurasi AWS Command Line Interface (AWS CLI) untuk menggunakan antarmuka baris perintah.

Daftar ke Amazon Web Services (AWS)

Ketika Anda membuat akun AWS, kami secara otomatis mendaftarkan akun Anda untuk semua layanan AWS. Anda hanya membayar untuk layanan yang digunakan.

Jika Anda sudah memiliki akun AWS, lewati ke langkah berikutnya. Jika Anda belum memiliki akun AWS, gunakan prosedur berikut untuk membuatnya.

Untuk mendaftar akun AWS

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

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

Masuk ke Konsol Amazon CloudWatch

Untuk masuk ke konsol Amazon CloudWatch

1. Masuk ke AWS Management Console dan buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Jika perlu, ubah wilayahnya. Dari bilah navigasi, pilih wilayah tempat sumber daya AWS Anda berada.
3. Di panel navigasi, pilih Peristiwa.

Kredensial Akun

Meskipun Anda dapat menggunakan kredensial pengguna root untuk mengakses CloudWatch Events, kami sarankan Anda menggunakan akun AWS Identity and Access Management (IAM). Jika Anda menggunakan akun IAM untuk mengakses CloudWatch, Anda harus memiliki izin berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "events:*",
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Siapkan Antarmuka Baris Perintah

Anda dapat menggunakan AWS CLI untuk menjalankan operasi CloudWatch Events.

Untuk informasi tentang cara memasang dan mengonfigurasi AWS CLI, lihat Menyiapkan AWS Command Line Interface di AWS Command Line Interface Panduan Pengguna.

Titik Akhir Regional

Anda harus mengaktifkan titik akhir wilayah (default) untuk menggunakan CloudWatch Events. Untuk informasi selengkapnya, lihat Mengaktifkan dan menonaktifkan AWS STS di AWS Wilayah dalam Panduan Pengguna IAM.

Memulai dengan Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Gunakan prosedur di bagian ini untuk membuat dan menghapus aturan CloudWatch Events. Ini adalah prosedur umum yang dapat digunakan untuk setiap sumber atau target peristiwa. Untuk tutorial mengenai skenario dan target tertentu, lihat Tutorials.

Semua aturan

Isi

- Membuat Aturan CloudWatch Events yang Memicu Peristiwa (p. 7)
- Membuat Aturan CloudWatch Events yang Memicu Panggilan API AWS Menggunakan AWS CloudTrail (p. 8)
- Membuat Aturan CloudWatch Events yang Memicu Jadwal (p. 9)
- Menghapus atau Menonaktifkan Aturan CloudWatch Events (p. 10)

Pembatasan

- Target yang Anda kaitkan dengan aturan harus berada di Wilayah yang sama dengan aturan.
- Beberapa tipe target mungkin tidak tersedia di semua wilayah. Untuk informasi selengkapnya, lihat Wilayah dan Titik akhir dalam Referensi Umum Amazon Web Services.
- Membuat aturan dengan target bawaan hanya didukung di AWS Management Console.
- Jika Anda membuat aturan dengan antrian Amazon SQS dienkripsi sebagai target, Anda harus memiliki bagian berikut yang disertakan dalam kebijakan kunci KMS Anda. Hal ini memungkinkan peristiwa berhasil dikirim ke antrian dienkripsi.

```
{
                "Sid": "Allow CWE to use the key",
                "Effect": "Allow",
                "Principal": {
                                "Service": "events.amazonaws.com"
                },
                "Action": [
                                "kms:Decrypt",
                                "kms:GenerateDataKey"
                ],
                "Resource": "*"
}
```

Membuat Aturan CloudWatch Events yang Memicu Peristiwa

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Terapkan langkah-langkah berikut untuk membuat aturan CloudWatch Events yang memicu peristiwa yang dikeluarkan oleh layanan AWS.

Untuk membuat aturan yang memicu peristiwa:

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Plih Pola Peristiwa, Bangun pola peristiwa agar cocok dengan peristiwa berdasarkan layanan.
   b. Untuk Nama Layanan, pilih layanan yang menghasilkan peristiwa untuk memicu aturan.
   c. Untuk Jenis Peristiwa, pilih peristiwa tertentu yang memicu aturan. Jika satu-satunya pilihan adalah Panggilan API AWS melalui CloudTrail, layanan yang dipilih tidak menghasilkan peristiwa dan Anda hanya dapat mendasarkan aturan pada API panggilan yang dibuat untuk layanan ini. Untuk informasi selengkapnya tentang pembuatan aturan ini, lihat Membuat Aturan CloudWatch Events yang Memicu Panggilan API AWS Menggunakan AWS CloudTrail (p. 8).
   d. Tergantung pada layanan yang menghasilkan peristiwa, Anda mungkin melihat opsi untuk Semua... dan ...Spesifik. Pilih Semua... untuk memiliki pemicu peristiwa pada semua jenis peristiwa yang dipilih, atau pilih ...Spesifik untuk memilih satu atau lebih jenis peristiwa tertentu.
4. Untuk Target, pilih Tambah Target dan pilih layanan AWS yang bertindak ketika suatu peristiwa dari jenis yang dipilih terdeteksi.
5. Di bidang lain di bagian ini, masukkan informasi khusus untuk jenis target ini, jika ada yang diperlukan.
6. Untuk banyak jenis target, CloudWatch Events membutuhkan izin untuk mengirim peristiwa ke target. Dalam kasus ini, CloudWatch Events dapat membuat IAM role yang diperlukan agar peristiwa Anda berjalan:
   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan IAM role yang Anda buat sebelumnya, pilih Gunakan peran yang sudah ada.
7. Secara opsional, ulangi langkah 4-6 untuk menambahkan target lain untuk aturan ini.
8. Pilih Konfigurasi detail. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.

   Nama aturan harus unik dalam Wilayah ini.
9. Pilih Buat aturan.

Membuat Aturan CloudWatch Events yang Memicu Panggilan API AWS Menggunakan AWS CloudTrail

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Untuk membuat aturan yang memicu tindakan oleh layanan AWS yang tidak menghasilkan peristiwa, Anda dapat mendasarkan aturan pada panggilan API yang dibuat oleh layanan tersebut. Panggilan API dicatat oleh AWS CloudTrail. Untuk informasi selengkapnya tentang panggilan API yang dapat Anda gunakan untuk aturan, lihat Layanan yang Didukung oleh Riwayat Peristiwa CloudTrail.

Aturan di CloudWatch Events hanya berfungsi di Wilayah tempat pembuatannya. Jika Anda mengonfigurasi CloudTrail untuk melacak panggilan API di beberapa Wilayah dan Anda menginginkan aturan berdasarkan CloudTrail untuk dipicu masing-masing Wilayah tersebut, Anda harus membuat aturan terpisah di setiap Wilayah yang ingin Anda lacak.

Semua peristiwa yang disampaikan oleh CloudTrail memiliki `AWS API Call via CloudTrail` sebagai nilai untuk `detail-type`.

> **Note**
>
> Di CloudWatch Events, aturan yang dibuat mungkin menyebabkan loop tak terbatas, di mana aturan dijalankan berulang kali. Misalnya, aturan mungkin mendeteksi bahwa ACL telah berubah di bucket S3, dan memicu perangkat lunak untuk mengubahnya ke keadaan yang diinginkan. Jika aturan tidak ditulis dengan hati-hati, perubahan berikutnya pada ACL akan mengaktifkan kembali aturan, yang membuat loop tak terbatas.
> Untuk mencegah hal ini, tulis aturan agar tindakan yang dipicu tidak mengaktifkan kembali aturan yang sama. Misalnya, aturan Anda hanya dapat berlaku jika keadaan ACL buruk, bukan setelah perubahan apa pun.
> Loop tak terbatas dapat dengan cepat mengakibatkan biaya yang lebih tinggi dari yang diperkirakan. Kami merekomendasikan agar Anda menggunakan penganggaran, yang akan memberi tahu Anda bila biaya melampaui batas yang ditentukan. Untuk informasi lebih lanjut, lihat Mengelola Biaya Anda dengan Anggaran.

Untuk membuat aturan yang memicu panggilan API melalui CloudTrail:

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Plih Pola Peristiwa, Bangun pola peristiwa agar cocok dengan peristiwa berdasarkan layanan.
   b. Untuk Nama Layanan, pilih layanan yang menggunakan operasi API untuk digunakan sebagai pemicu.
   c. Untuk Jenis acara, pilih Panggilan API AWS melalui CloudTrail.
   d. Untuk memicu aturan Anda ketika setiap operasi API untuk layanan ini dipanggil, pilih Setiap operasi. Untuk memicu aturan Anda hanya ketika operasi API tertentu dipanggil, pilih Operasi spesifik, ketik nama operasi di kotak berikutnya, dan kemudian tekan ENTER. Untuk menambahkan lebih banyak operasi, pilih +.

4. Untuk Target, pilih Tambah Target dan pilih layanan AWS yang bertindak ketika suatu peristiwa dari jenis yang dipilih terdeteksi.
5. Di bidang lain di bagian ini, masukkan informasi khusus untuk jenis target ini, jika ada yang diperlukan.
6. Untuk banyak jenis target, CloudWatch Events membutuhkan izin untuk mengirim peristiwa ke target. Dalam kasus ini, CloudWatch Events dapat membuat IAM role yang diperlukan agar peristiwa Anda berjalan:

   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan IAM role yang Anda buat sebelumnya, pilih Gunakan peran yang sudah ada.
7. Secara opsional, ulangi langkah 4-6 untuk menambahkan target lain untuk aturan ini.
8. Pilih Konfigurasi detail. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.

   Nama aturan harus unik dalam Wilayah ini.
9. Pilih Buat aturan.

Membuat Aturan CloudWatch Events yang Memicu Jadwal

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Pelajari cara membuat aturan CloudWatch Events yang memicu jadwal reguler.

Membuat aturan yang memicu jadwal reguler

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, pilih Jadwalkan.
4. Pilih Tingkat tetap dan tentukan seberapa sering tugas berjalan, atau pilih Ekspresi cron dan tentukan ekspresi cron yang menentukan kapan tugas tersebut akan dipicu. Untuk informasi selengkapnya tentang sintaks ekspresi cron, lihat Ekspresi Jadwal untuk Aturan (p. 33).
5. Untuk Target, pilih Tambah Target dan pilih layanan AWS yang bertindak ketika suatu peristiwa dari jenis yang dipilih terdeteksi.
6. Di bidang lain di bagian ini, masukkan informasi khusus untuk jenis target ini, jika ada yang diperlukan.
7. Untuk banyak jenis target, CloudWatch Events membutuhkan izin untuk mengirim peristiwa ke target. Dalam kasus ini, CloudWatch Events dapat membuat IAM role yang diperlukan agar peristiwa Anda berjalan:

   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan IAM role yang Anda buat sebelumnya, pilih Gunakan peran yang ada.
8. Secara opsional, ulangi langkah 5-7 untuk menambahkan target lain untuk aturan ini.
9. Pilih Konfigurasi detail. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.

   Nama aturan harus unik dalam Wilayah ini.
10. Pilih Buat aturan.

Menghapus atau Menonaktifkan Aturan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Gunakan langkah-langkah berikut untuk menghapus atau menonaktifkan aturan CloudWatch Events.

Menghapus atau menonaktifkan aturan

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Aturan.

   Aturan terkelola memiliki ikon kotak di samping namanya. Untuk informasi selengkapnya, lihat Aturan Terkelola Amazon CloudWatch Events (p. 98).
3. Lakukan salah satu dari berikut:

   a. Untuk menghapus aturan, pilih tombol di samping aturan dan pilih Tindakan, Hapus, Hapus.

      Jika aturan adalah aturan terkelola, masukkan nama aturan untuk mengakui bahwa aturan itu adalah aturan terkelola dan bahwa menghapusnya dapat menghentikan fungsionalitas dalam layanan yang membuat aturan. Untuk melanjutkan, masukkan nama aturan dan pilih Hapus paksa.

   b. Untuk menonaktifkan aturan sementara, pilih tombol di samping aturan dan pilih Tindakan, Nonaktifkan, Nonaktifkan.

      Anda tidak dapat menonaktifkan aturan terkelola.

Tutorial CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Tutorial berikut menunjukkan cara membuat aturan CloudWatch Events untuk tugas dan target tertentu.

Tutorial:

- Tutorial: Menggunakan CloudWatch EventsAWS Systems ManagerRun Command (p. 11)
- Tutorial: Mencatat Status Instans Amazon EC2 Menggunakan CloudWatch Events (p. 12)
- Tutorial: Mencatat Status Grup Auto Scaling Menggunakan CloudWatch Events (p. 14)
- Tutorial: Catat Operasi Tingkat Objek Amazon S3 Menggunakan CloudWatch Events (p. 16)
- Tutorial: Gunakan Transformator Input untuk Menyesuaikan Apa yang Diteruskan ke Target Peristiwa (p. 19)
- Tutorial: LogAWSPanggilan API Menggunakan CloudWatch Events (p. 20)
- Tutorial: Jadwalkan Snapshot Amazon EBS Otomatis Menggunakan CloudWatch Events (p. 22)
- Tutorial: JadwalAWS LambdaFungsi Menggunakan CloudWatch Events (p. 23)
- Tutorial: SETAWS Systems ManagerOtomatisasi sebagai Target CloudWatch Events (p. 26)
- Tutorial: Relai Peristiwa Amazon Kinesis Stream Menggunakan CloudWatch Events (p. 27)
- Tutorial: Jalankan Tugas Amazon ECS ketika File Diunggah ke Bucket Amazon S3 (p. 29)
- Tutorial: Jadwalkan Pembangunan Otomatis Menggunakan CodeBuild (p. 30)
- Tutorial: Mencatat Perubahan Status Instans Amazon EC2 (p. 31)

Tutorial: Menggunakan CloudWatch EventsAWS Systems ManagerRun Command

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menggunakan Amazon CloudWatch Events untuk memanggil Run Command AWS Systems Manager dan melakukan tindakan pada instans Amazon EC2 ketika peristiwa spesifik terjadi. Dalam tutorial

ini, siapkan Run Command Systems Manager untuk menjalankan perintah shell dan mengonfigurasi setiap instans baru yang diluncurkan dalam grup Amazon EC2 Auto Scaling. Tutorial ini mengasumsikan bahwa Anda telah diberi sebuah tag untuk grup Amazon EC2 Auto Scaling, dengan `environment` sebagai kunci dan `production` sebagai nilai.

Untuk membuat aturan CloudWatch Events

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Plih Pola Peristiwa, Bangun pola peristiwa agar cocok dengan peristiwa berdasarkan layanan.
   b. Untuk Nama Layanan, pilih Auto Scaling. Untuk Jenis peristiwa, pilih Peluncuran dan Penghentian Instans.
   c. PilihPeristiwa khusus instans, Tindakan Siklus Hidup Peluncuran Instans EC2.
   d. Secara default, aturan cocok dengan grup Amazon EC2 Auto Scaling di wilayah tersebut. Untuk membuat aturan cocok dengan grup tertentu, pilih Nama grup spesifik lalu pilih satu atau lebih grup.
4. Untuk Target, pilih Tambah Target, Run Command SSM.
5. Untuk Dokumen, pilih AWS-RunshellScript (Linux). Ada banyak pilihan Dokumen lainnya yang mencakup instans Linux dan Windows. Untuk jenis Kunci target, ketik **`tag:environment`**. Untuk Nilai target, ketik **`production`**, dan pilih Tambahkan.
6. Di bawah Konfigurasi parameter, pilih Konstanta.
7. Untuk Perintah, ketik perintah shell dan pilih Tambahkan. Ulangi langkah ini untuk semua perintah yang ingin dijalankan saat sebuah instans diluncurkan.
8. Jika perlu, ketik informasi yang sesuai di WorkingDirectory dan ExecutionTimeout.
9. CloudWatch Events dapat membuat IAM role yang diperlukan agar peristiwa Anda berjalan:
   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan IAM role yang Anda buat sebelumnya, pilih Gunakan peran yang sudah ada.
10. Pilih Konfigurasi detail. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
11. Pilih Buat aturan.

Tutorial: Mencatat Status Instans Amazon EC2 Menggunakan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat membuat fungsi AWS Lambda yang mencatat perubahan status untuk instans Amazon EC2. Anda dapat memilih untuk membuat aturan yang menjalankan fungsi setiap kali terjadi transisi status atau transisi ke satu atau beberapa status yang menarik. Dalam tutorial ini, Anda log peluncuran setiap instans baru.

Langkah 1: BuatAWS LambdaFungsi

Buat fungsi Lambda untuk log peristiwa perubahan status. Anda menetapkan fungsi ini saat membuat aturan Anda.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol tersebut di https://console.aws.amazon.com/lambda/.
2. Jika Anda baru mengenal Lambda, Anda akan melihat halaman selamat datang. Pilih Mulai Sekarang. Atau, pilih Buat fungsi Lambda.
3. Pada halaman Pilih cetak biru, ketik `hello` untuk filter, lalu pilih cetak biru hello-world.
4. Pada halaman Mengonfigurasi pemicu, pilih Selanjutnya.
5. Pada halaman Konfigurasi fungsi, lakukan hal berikut:
   a. Ketik nama dan deskripsi untuk fungsi Lambda. Misalnya, beri nama fungsi “LogEC2InstanceStateChange”.
   b. Edit kode sampel untuk fungsi Lambda. Misalnya:

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2InstanceStateChange');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. Untuk Peran, pilih Pilih peran yang ada. Untuk Peran yang ada, pilih peran eksekusi dasar Anda. Atau, buat peran eksekusi dasar.
   d. Pilih Selanjutnya.
6. Pada halaman Peninjauan, pilih Buat fungsi.

Langkah 2: Buat aturan

Buat aturan untuk menjalankan fungsi Lambda Anda setiap kali Anda meluncurkan instans Amazon EC2.

Untuk membuat aturan CloudWatch Events

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola kejadian.
   b. Pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   c. Pilih EC2, Notifikasi Perubahan Status Instans EC2.
   d. Pilih Status khusus, Berjalan.
   e. Secara default, aturan cocok dengan semua instans di wilayah tersebut. Untuk membuat aturan cocok dengan instans tertentu, pilih Instans tertentu dan pilih satu atau beberapa instans.
4. UntukTarget, pilih Menambahkan target, Fungsi Lambda.
5. Untuk Fungsi, pilih fungsi Lambda yang Anda buat.
6. Pilih Konfigurasi detail.
7. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.

8. Pilih Buat aturan.

Langkah 3: Uji Aturan

Untuk menguji aturan Anda, luncurkan instans Amazon EC2. Setelah menunggu beberapa menit hingga instans diluncurkan dan diinisialisasi, Anda dapat memverifikasi bahwa fungsi Lambda Anda telah dipanggil.

Untuk menguji aturan Anda dengan meluncurkan instans

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Peluncuran instans. Untuk informasi selengkapnya, lihat Peluncuran Instans Anda di Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
4. Di panel navigasi, pilih Peristiwa, Aturan, lalu pilih nama aturan yang Anda buat, dan kemudian pilih Tampilkan metrik untuk aturan tersebut.
5. Untuk melihat output dari fungsi Lambda Anda, lakukan hal berikut:
   a. Di panel navigasi, pilih Log.
   b. Pilih nama grup log untuk fungsi Lambda Anda (/aws/lambda/function-name).
   c. Pilih nama aliran log untuk melihat data yang disediakan oleh fungsi untuk instans yang Anda luncurkan.
6. (Opsional) Setelah selesai, Anda dapat membuka konsol Amazon EC2 dan menghentikan atau mengakhiri instans yang Anda luncurkan. Untuk informasi lebih lanjut, lihat Akhiri Instans Anda di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Tutorial: Mencatat Status Grup Auto Scaling Menggunakan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menjalankan fungsi AWS Lambda yang mencatat peristiwa ketika grup Auto Scaling meluncurkan atau menghentikan instans Amazon EC2 dan mencatat apakah peluncuran atau penghentian tersebut berhasil.

Untuk informasi tentang skenario CloudWatch Events lainnya yang menggunakan peristiwa Amazon EC2 Auto Scaling, lihat Mendapatkan CloudWatch Events Saat Grup Auto Scaling Menskalakan dalam Panduan Pengguna Amazon EC2 Auto Scaling.

Langkah 1: BuatAWS LambdaFungsi

Buat fungsi Lambda untuk log menskalakan keluar dan menskalaan kedalam peristiwa untuk grup Auto Scaling Anda. Anda menetapkan fungsi ini saat membuat aturan Anda.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol tersebut di https://console.aws.amazon.com/lambda/.
2. Jika Anda baru mengenal Lambda, Anda akan melihat halaman selamat datang. Pilih Mulai Sekarang. Atau, pilih Buat fungsi Lambda.
3. Pada halaman Pilih cetak biru, ketik `hello` untuk filter, lalu pilih cetak biru hello-world.
4. Pada halaman Mengonfigurasi pemicu, pilih Selanjutnya.
5. Pada halaman Konfigurasi fungsi, lakukan hal berikut:
   a. Ketik nama dan deskripsi untuk fungsi Lambda. Misalnya, beri nama fungsi “LogAutoScalingEvent”.
   b. Edit kode sampel untuk fungsi Lambda. Misalnya:

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogAutoScalingEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. Untuk Peran, pilih Pilih peran yang ada. Untuk Peran yang ada, pilih peran eksekusi dasar Anda. Atau, buat peran eksekusi dasar.
   d. Pilih Selanjutnya.
6. Pilih Buat fungsi.

Langkah 2: Buat aturan

Buat aturan untuk menjalankan fungsi Lambda Anda setiap kali grup Auto Scaling Anda meluncurkan atau menghentikan instans.

Untuk membuat tabel

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola kejadian.
   b. Pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   c. Pilih Auto Scaling, Peluncuran dan Penghentian Instans.
   d. Untuk merekam semua peristiwa peluncuran dan penghentian instans yang berhasil dan tidak berhasil, pilih Peristiwa instans apa pun.
4. Secara default, aturan cocok dengan grup Auto Scaling di Wilayah. Untuk membuat aturan cocok dengan grup Auto Scaling tertentu, pilih Nama grup tertentu, lalu pilih satu atau beberapa grup Auto Scaling.
5. UntukTarget, pilih Menambahkan target, Fungsi Lambda.
6. Untuk Fungsi, pilih fungsi Lambda yang Anda buat.
7. Pilih Konfigurasi detail.
8. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan. Misalnya, jelaskan aturan sebagai “Catat setiap kali grup Auto Scaling menskalakan keluar atau ke dalam”.
9. Pilih Buat aturan.

Langkah 3: Uji Aturan

Anda dapat menguji aturan Anda dengan penskalaan grup Auto Scaling secara manual sehingga meluncurkan sebuah instans. Tunggu beberapa menit hingga peristiwa penskalaan keluar terjadi, lalu verifikasi bahwa fungsi Lambda Anda telah dipanggil.

Untuk menguji aturan menggunakan grup Auto Scaling

1. Untuk menambah ukuran grup Auto Scaling Anda, lakukan hal berikut:
   a. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
   b. Pada panel navigasi, pilih Auto Scaling, Grup Auto Scaling.
   c. Pilih kotak centang di samping grup Auto Scaling Anda.
   d. Pada tab Detail, pilih Edit. Untuk Diinginkan, tingkatkan kapasitas yang diinginkan sebesar satu. Misalnya, jika nilai saat ini adalah 2, ketik 3. Kapasitas yang diinginkan harus kurang dari atau sama dengan ukuran maksimum grup. Jika nilai baru Anda untuk Diinginkan lebih besar dari Maks, Anda harus memperbarui Maks. Setelah selesai, pilih Simpan.
2. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
3. Di panel navigasi, pilih Peristiwa, Aturan, lalu pilih nama aturan yang Anda buat, dan kemudian pilih Tampilkan metrik untuk aturan tersebut.
4. Untuk melihat output dari fungsi Lambda Anda, lakukan hal berikut:
   a. Di panel navigasi, pilih Log.
   b. Pilih nama grup log untuk fungsi Lambda Anda (aws/lambda/function-name).
   c. Pilih nama aliran catatan untuk menampilkan data yang disediakan oleh fungsi untuk instans yang Anda luncurkan.
5. (Opsional) Setelah selesai, Anda dapat mengurangi kapasitas yang diinginkan sebesar satu sehingga grup Auto Scaling kembali ke ukuran sebelumnya.

Tutorial: Catat Operasi Tingkat Objek Amazon S3 Menggunakan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat mencatat operasi API tingkat objek pada bucket S3 Anda. Sebelum Amazon CloudWatch Events dapat mencocokkan peristiwa, Anda harus menggunakan AWS CloudTrail untuk menyiapkan dan mengonfigurasi jejak untuk menerima peristiwa ini.

Langkah 1: Konfigurasi AndaAWS CloudTrailTrail

Untuk mencatat peristiwa data untuk bucket S3 ke AWS CloudTrail dan CloudWatch Events, buat sebuah jejak. Jejak merekam panggilan API dan peristiwa terkait di akun Anda, kemudian mengirimkan berkas log ke bucket S3 yang Anda tentukan. Anda dapat memperbarui jejak yang ada atau membuat jejak baru.

Untuk membuat jejak

1. Buka konsol CloudTrail di https://console.aws.amazon.com/cloudtrail/.
2. Di panel navigasi, pilih Jejak, Buat jejak.
3. Untuk Nama jejak, ketikkan nama untuk jejak.
4. Untuk Peristiwa data, ketik nama bucket dan prefiks (opsional). Untuk setiap jejak, Anda dapat menambahkan hingga 250 objek Amazon S3.
   - Untuk mencatat kejadian data untuk semua objek Amazon S3 dalam bucket, tentukan bucket S3 dan prefiks kosong. Ketika suatu kejadian terjadi pada sebuah objek di bucket tersebut, jejak meproses dan mencatat kejadian.
   - Untuk mencatat peristiwa data untuk objek Amazon S3 tertentu, pilih Tambahkan bucket S3, lalu tentukan bucket S3 dan boleh juga prefiks objeknya. Ketika suatu kejadian terjadi pada sebuah objek di bucket tersebut dan objek dimulai dengan prefiks yang ditentukan, jejak memproses dan mencatat kejadian tersebut.
5. Untuk setiap sumber daya, tentukan apakah Anda akan mencatat peristiwa Baca, peristiwa Tulis, atau keduanya.
6. Untuk Lokasi penyimpanan, buat atau pilih bucket S3 yang ada sebagai penyimpanan berkas log yang ditentukan.
7. Pilih Create (Buat).

Untuk informasi selengkapnya, lihat Peristiwa Data dalam Panduan Pengguna AWS CloudTrail.

Langkah 2: BuatAWS LambdaFungsi

Buat fungsi Lambda untuk mencatat kejadian data untuk bucket S3 Anda. Anda menetapkan fungsi ini saat membuat aturan Anda.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol tersebut di https://console.aws.amazon.com/lambda/.
2. Jika Anda baru mengenal Lambda, Anda akan melihat halaman selamat datang. Pilih Buat fungsi. Atau, pilih Buat fungsi.
3. Pilih Tulis dari awal.
4. Di bawah Tulis dari awal, lakukan langkah berikut:
   a. Ketik nama untuk fungsi Lambda. Misalnya, beri nama fungsi “LogS3DataEvents”.
   b. Untuk Peran, pilih Buat peran kustom.

      Jendela baru terbuka. Ubah Nama peran jika perlu, dan pilih Izinkan.
   c. Kembali ke konsol Lambda, pilih Buat fungsi.
5. Edit kode untuk fungsi Lambda menjadi kode berikut, lalu pilih Simpan.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogS3DataEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

Langkah 3: Buat aturan

Buat aturan untuk menjalankan fungsi Lambda Anda terkait peristiwa data Amazon S3.

Untuk membuat tabel

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Aturan, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola kejadian.
   b. Pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   c. Pilih Simple Storage Service (S3), Operasi Tingkat Objek.
   d. Pilih Operasi spesifik, PutObject.
   e. Secara default, aturan sesuai dengan peristiwa data untuk semua bucket di wilayah tersebut. Untuk mencocokkan peristiwa data untuk bucket tertentu, pilih Tentukan bucket berdasarkan nama lalu tentukan satu atau beberapa bucket.
4. UntukTarget, pilih Menambahkan target, Fungsi Lambda.
5. Untuk Fungsi, pilih fungsi Lambda yang Anda buat.
6. Pilih Konfigurasi detail.
7. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
8. Pilih Buat aturan.

Langkah 4: Uji Aturan

Untuk menguji aturan, letakkan objek di bucket S3 Anda. Anda dapat memverifikasi bahwa fungsi Lambda Anda dipanggil.

Untuk menampilkan catatan fungsi Lambda Anda

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log.
3. Pilih nama grup log untuk fungsi Lambda Anda (aws/lambda/function-name).
4. Pilih nama aliran catatan untuk menampilkan data yang disediakan oleh fungsi untuk instans yang Anda luncurkan.

Anda juga dapat memeriksa isi log CloudTrail Anda di bucket S3 yang Anda tentukan untuk jejak Anda. Untuk informasi selengkapnya, lihat Mendapatkan dan Melihat CloudTrail Log Files Anda di AWS CloudTrail Panduan Pengguna.

Tutorial: Gunakan Transformator Input untuk Menyesuaikan Apa yang Diteruskan ke Target Peristiwa

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menggunakan fitur transformator input CloudWatch Events untuk menyesuaikan teks dari peristiwa sebelum diinput ke target aturan.

Anda dapat menentukan beberapa jalur JSON dari peristiwa tersebut dan menetapkan outputnya ke variabel yang berbeda. Kemudian Anda dapat menggunakan variabel tersebut di templat input sebagai <*variable-name*>. Karakter < dan > tidak dapat lepas.

Jika Anda menentukan variabel untuk mencocokkan jalur JSON yang tidak ada dalam peristiwa, maka variabel tersebut tidak akan dibuat dan tidak muncul dalam output.

Dalam tutorial ini, kita mengekstraksi id-instans dan status instans Amazon EC2 dari peristiwa perubahan status instans. Kami menggunakan transformator input untuk memasukkan data ke dalam pesan yang mudah dibaca yang dikirim ke topik Amazon SNS. Aturan ini akan dipicu ketika instans mana pun mengalami perubahan status. Misalnya, dengan aturan ini, peristiwa notifikasi perubahan status instans Amazon EC2 berikut menghasilkan pesan Amazon SNS berupa Instans EC2 i-1234567890abcdef0 telah berubah status menjadi berhenti.

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2015-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/ i-1234567890abcdef0"
   ],
   "detail":{
      "instance-id":" i-1234567890abcdef0",
      "state":"stopped"
   }
}
```

Kami melakukannya dengan memetakan variabel *instans* ke jalur JSON `$.detail.instance-id` dari peristiwa tersebut, dan memetakan variabel *status* ke jalur JSON `$.detail.state`. Kemudian, kami mengatur templat input sebagai "Instans EC2 <instance> telah berubah status menjadi <state>."

Buat aturan

Untuk menyesuaikan informasi perubahan status instans yang dikirimkan ke target menggunakan transformator input

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.

2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola kejadian.
   b. Pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   c. Pilih EC2, Notifikasi Perubahan Status Instans EC2.
   d. Pilih Status apa pun, Semua instans.
4. Untuk Target, pilih Tambah target, Topik SNS.
5. Untuk Topik, pilih topik Amazon SNS yang akan diberitahukan ketika instans Amazon EC2 berubah statusnya.
6. Pilih Mengkonfigurasi input, Transformer input.
7. Di kotak berikutnya, ketik {"state" : “$.detail.state”, “instance” : “$.detail.instance-id"}
8. Di kotak berikut ini, ketik “Instans EC2 <instance> telah berubah status menjadi <state>.”
9. Pilih Konfigurasikan detail.
10. Ketikkan sebuah nama dan deskripsi untuk aturan, lalu pilih Buat aturan.

Tutorial: LogAWSPanggilan API Menggunakan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menggunakan fungsi AWS Lambda yang mencatat setiap panggilan API AWS. Misalnya, Anda dapat membuat aturan untuk mencatat setiap operasi di Amazon EC2, atau Anda dapat membatasi aturan ini untuk hanya mencatat panggilan API tertentu. Dalam tutorial ini, Anda akan mencatat setiap kali instans Amazon EC2 dihentikan.

Prasyarat

Sebelum Anda dapat mencocokkan peristiwa ini, Anda harus menggunakan AWS CloudTrail untuk menyiapkan jejak. Jika Anda tidak memiliki jejak, selesaikan prosedur berikut.

Untuk membuat jejak

1. Buka konsol CloudTrail di https://console.aws.amazon.com/cloudtrail/.
2. Pilih Jejak, Buat jejak.
3. Untuk Nama jejak, ketikkan nama untuk jejak.
4. Untuk Lokasi penyimpanan, pada Buat bucket S3 baru, ketik nama untuk bucket baru yang akan dikirimi log oleh CloudTrail.
5. Pilih Create (Buat).

Langkah 1: BuatAWS LambdaFungsi

Buat fungsi Lambda untuk log peristiwa panggilan API. Anda menetapkan fungsi ini saat membuat aturan Anda.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol tersebut di https://console.aws.amazon.com/lambda/.
2. Jika Anda baru mengenal Lambda, Anda akan melihat halaman selamat datang. Pilih Mulai Sekarang. Atau, pilih Buat fungsi Lambda.
3. Pada halaman Pilih cetak biru, ketik `hello` untuk filter, lalu pilih cetak biru hello-world.
4. Pada halaman Mengonfigurasi pemicu, pilih Selanjutnya.
5. Pada halaman Konfigurasi fungsi, lakukan hal berikut:
   a. Ketik nama dan deskripsi untuk fungsi Lambda. Misalnya, beri nama fungsi “LogEC2StopInstance”.
   b. Edit kode sampel untuk fungsi Lambda. Misalnya:

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2StopInstance');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. Untuk Peran, pilih Pilih peran yang ada. Untuk Peran yang ada, pilih peran eksekusi dasar Anda. Atau, buat peran eksekusi dasar.
   d. Pilih Selanjutnya.
6. Pada halaman Peninjauan, pilih Buat fungsi.

Langkah 2: Buat aturan

Buat aturan untuk menjalankan fungsi Lambda setiap kali Anda menghentikan instans Amazon EC2.

Untuk membuat tabel

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola kejadian.
   b. Pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   c. Pilih EC2, Panggilan API AWS melalui CloudTrail.
   d. Pilih Operasi spesifik dan ketik `StopInstances` pada kotak di bawah ini.
4. UntukTarget, pilih Menambahkan target, Fungsi Lambda.
5. Untuk Fungsi, pilih fungsi Lambda yang Anda buat.
6. Pilih Konfigurasi detail.
7. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
8. Pilih Buat aturan.

Langkah 3: Uji Aturan

Anda dapat menguji aturan Anda dengan pemfilteran stopword instans Amazon EC2 menggunakan konsol Amazon EC2. Tunggu beberapa menit hingga instans berhenti, lalu periksa metrik AWS Lambda Anda di konsol CloudWatch untuk memverifikasi bahwa fungsi Anda telah dipanggil.

Untuk menguji aturan Anda dengan pemfilteran stopword instans

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Peluncuran instans. Untuk informasi selengkapnya, lihat Peluncuran Instans Anda di Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Hentikan instans. Untuk informasi lebih lanjut, lihat Hentikan dan Mulai Instans Anda dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
4. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
5. Di panel navigasi, pilih Peristiwa, pilih nama aturan yang Anda buat, dan kemudian pilih Tampilkan metrik untuk aturan tersebut.
6. Untuk melihat output dari fungsi Lambda Anda, lakukan hal berikut:
   a. Di panel navigasi, pilih Log.
   b. Pilih nama grup log untuk fungsi Lambda Anda (aws/lambda/function-name).
   c. Pilih nama aliran log untuk melihat data yang disediakan oleh fungsi untuk instans yang Anda hentikan.
7. (Opsional) Ketika selesai, Anda dapat mengakhiri instans yang dihentikan. Untuk informasi lebih lanjut, lihat Akhiri Instans Anda di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Tutorial: Jadwalkan Snapshot Amazon EBS Otomatis Menggunakan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menjalankan aturan CloudWatch Events sesuai jadwal. Dalam tutorial ini, Anda membuat snapshot volume dari Amazon Elastic Block Store (Amazon EBS) yang ada secara otomatis sesuai jadwal. Anda dapat memilih nilai tetap untuk membuat snapshot setiap beberapa menit atau menggunakan ekspresi cron untuk menentukan jadwal pembuatan snapshot pada waktu tertentu dalam sehari.

> **Important**
>
> Membuat aturan dengan target bawaan hanya didukung di AWS Management Console.

Langkah 1: Buat aturan

Buat aturan yang mengambil snapshot pada jadwal. Anda dapat menggunakan ekspresi nilai atau ekspresi cron untuk menentukan jadwal. Selengkapnya, lihat Ekspresi Jadwal untuk Aturan (p. 33).

Untuk membuat aturan

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber Peristiwa, lakukan hal berikut:
   a. Pilih Jadwal.
   b. Pilih Nilai tetap dan tentukan interval jadwal (misalnya, 5 menit). Atau, pilih Ekspresi cron dan tentukan ekspresi cron (misalnya, setiap 15 menit, Senin sampai Jumat, mulai dari waktu saat ini).
4. Untuk Target, pilih Tambahkan target, lalu pilih Panggilan API EC2 CreateSnapshot. Anda mungkin harus menggulir ke atas dalam daftar kemungkinan target untuk menemukan Panggilan API EC2 CreateSnapshot.
5. Untuk ID Volume, ketik ID volume dari volume Amazon EBS yang ditargetkan.
6. Pilih Membuat peran baru untuk sumber daya khusus ini. Peran baru memberikan izin target untuk mengakses sumber daya atas nama Anda.
7. Pilih Konfigurasi detail.
8. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
9. Pilih Buat aturan.

Langkah 2: Uji Aturan

Anda dapat memverifikasi aturan dengan melihat snapshot pertama Anda setelah diambil.

Untuk menguji aturan Anda

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, di bagian Elastic Block Store, pilih Snapshot.
3. Verifikasi bahwa snapshot pertama muncul di daftar.
4. (Opsional) Setelah selesai, Anda dapat menonaktifkan aturan untuk mencegah pengambilan snapshot tambahan.
   a. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
   b. Di panel navigasi, pilih Peristiwa, Aturan.
   c. Pilih aturan kemudian pilih Tindakan, Nonaktifkan.
   d. Ketika dimintai konfirmasi, pilih Nonaktifkan.

Tutorial: JadwalAWS LambdaFungsi Menggunakan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat mengatur Aturan untuk menjalankanAWS Lambdaberfungsi pada jadwal. Tutorial ini menunjukkan cara menggunakan AWS Management Console atau AWS CLI atau membuat aturan. Jika

Anda ingin menggunakan AWS CLI tapi belum menginstalnya, lihat Panduan Pengguna AWS Command Line Interface.

CloudWatch Events tidak memberikan presisi tingkat kedua dalam ekspresi jadwal. Resolusi terbaik untuk penggunaan ekspresi cron adalah satu menit. Karena CloudWatch Events dan layanan target terdistribusi, dapat terjadi penundaan beberapa detik antara waktu aturan yang dijadwalkan dipicu dan waktu layanan target menjalankan sumber daya target. Aturan terjadwal Anda akan dipicu di menit itu tetapi tidak persis pada detik ke-0.

Langkah 1: BuatAWS LambdaFungsi

Buat fungsi Lambda untuk mencatat peristiwa yang dijadwalkan. Anda menetapkan fungsi ini saat membuat aturan Anda.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol tersebut di https://console.aws.amazon.com/lambda/.
2. Jika Anda baru mengenal Lambda, Anda akan melihat halaman selamat datang. Pilih Mulai Sekarang. Atau, pilih Buat fungsi Lambda.
3. Pada halaman Pilih cetak biru, ketik `hello` untuk filter, lalu pilih cetak biru hello-world.
4. Pada halaman Mengonfigurasi pemicu, pilih Selanjutnya.
5. Pada halaman Konfigurasi fungsi, lakukan hal berikut:
   a. Ketik nama dan deskripsi untuk fungsi Lambda. Misalnya, beri nama fungsi “LogScheduledEvent”.
   b. Edit kode sampel untuk fungsi Lambda. Misalnya:

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. Untuk Peran, pilih Pilih peran yang ada. Untuk Peran yang ada, pilih peran eksekusi dasar Anda. Atau, buat peran eksekusi dasar.
   d. Pilih Selanjutnya.
6. Pada halaman Peninjauan, pilih Buat fungsi.

Langkah 2: Buat aturan

Buat aturan untuk menjalankan fungsi Lambda sesuai jadwal.

Untuk membuat aturan menggunakan konsol

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber Peristiwa, lakukan hal berikut:
   a. Pilih Jadwal.
   b. Pilih Nilai tetap dan tentukan interval jadwal (misalnya, 5 menit).
4. UntukTarget, pilih Menambahkan target, Fungsi Lambda.
5. Untuk Fungsi, pilih fungsi Lambda yang Anda buat.

6. Pilih Konfigurasi detail.
7. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
8. Pilih Buat aturan.

Jika mau, Anda dapat membuat aturan menggunakan AWS CLI. Pertama, Anda harus memberikan izin aturan untuk mengaktifkan fungsi Lambda Anda. Kemudian Anda dapat membuat aturan dan menambahkan fungsi Lambda sebagai target.

Untuk membuat topik menggunakan AWS CLI

1. Gunakan perintah berikut put-aturan untuk membuat aturan yang terpicu otomatis sesuai jadwal:

```
aws events put-rule \
--name my-scheduled-rule \
--schedule-expression 'rate(5 minutes)'
```

   Ketika aturan ini terpicu, akan dihasilkan sebuah peristiwa yang berfungsi sebagai input untuk target aturan ini. Berikut ini adalah contoh peristiwa:

```
{
    "version": "0",
    "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
    "detail-type": "Scheduled Event",
    "source": "aws.events",
    "account": "123456789012",
    "time": "2015-10-08T16:53:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule"
    ],
    "detail": {}
}
```

2. Gunakan perintah berikut add-permission untuk memercayakan prinsipal layanan CloudWatch Events (events.amazonaws.com) dan izin lingkup pada aturan dengan Amazon Resource Name (ARN) yang ditentukan:

```
aws lambda add-permission \
--function-name LogScheduledEvent \
--statement-id my-scheduled-event \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule
```

3. Gunakan perintah berikut put-targets untuk menambahkan fungsi Lambda yang Anda buat untuk aturan ini supaya fungsi ini berjalan setiap lima menit:

```
aws events put-targets --rule my-scheduled-rule --targets file://targets.json
```

   Buat file `targets.json` dengan isi sebagai berikut:

```
[
  {
    "Id": "1",
    "Arn": "arn:aws:lambda:us-east-1:123456789012:function:LogScheduledEvent"
  }
]
```

Langkah 3: Verifikasi Aturan

Setidaknya lima menit setelah menyelesaikan Langkah 2, Anda dapat memverifikasi bahwa fungsi Lambda Anda telah dipanggil.

Menguji aturan Anda

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Aturan, lalu pilih nama aturan yang Anda buat, dan kemudian pilih Tampilkan metrik untuk aturan tersebut.
3. Untuk melihat output dari fungsi Lambda Anda, lakukan hal berikut:
   a. Di panel navigasi, pilih Log.
   b. Pilih nama grup log untuk fungsi Lambda Anda (aws/lambda/function-name).
   c. Pilih nama aliran catatan untuk menampilkan data yang disediakan oleh fungsi untuk instans yang Anda luncurkan.
4. (Opsional) Setelah selesai, Anda dapat menonaktifkan aturan.
   a. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
   b. Di panel navigasi, pilih Peristiwa, Aturan.
   c. Pilih aturan kemudian pilih Tindakan, Nonaktifkan.
   d. Ketika dimintai konfirmasi, pilih Nonaktifkan.

Tutorial: SETAWS Systems ManagerOtomatisasi sebagai Target CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menggunakan CloudWatch Events untuk menjalankan Otomatisasi AWS Systems Manager pada jadwal waktu biasa atau ketika peristiwa tertentu terdeteksi. Tutorial ini mengasumsikan bahwa Anda menjalankan Otomatisasi Systems Manager berdasarkan peristiwa tertentu.

Untuk membuat aturan CloudWatch Events

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola Peristiwa kemudian pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   b. Untuk Nama Layanan dan Jenis Peristiwa, pilih jenis layanan dan jenis peristiwa yang akan digunakan sebagai pemicu.

      Tergantung pada layanan dan jenis peristiwa yang Anda pilih, Anda mungkin perlu menentukan opsi tambahan pada bagian Sumber Peristiwa.

4. Untuk Target, pilih Tambahkan Target, Otomatisasi SSM.
5. Untuk Dokumen, pilih dokumen Systems Manager yang akan dijalankan ketika target dipicu.
6. (Opsional) Untuk menentukan versi tertentu dari dokumen, pilih Konfigurasikan versi dokumen.
7. Di bawah Konfigurasikan parameter, pilih Tanpa Parameter atau Konstanta.

   Jika Anda memilihKonstan, tentukan konstanta yang akan diteruskan ke eksekusi dokumen.
8. CloudWatch Events dapat membuat IAM role yang diperlukan agar peristiwa Anda berjalan:

   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan IAM role yang Anda buat sebelumnya, pilih Gunakan peran yang sudah ada.
9. Pilih Konfigurasi detail. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
10. Pilih Buat aturan.

Tutorial: Relai Peristiwa Amazon Kinesis Stream Menggunakan CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat merelai peristiwa panggilan API AWS di CloudWatch Events ke aliran di Amazon Kinesis.

Prasyarat

Instal AWS CLI. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Command Line Interface.

Langkah 1: Membuat Amazon Kinesis Stream

Gunakan perintah berikut create-stream untuk membuat aliran.

```
aws kinesis create-stream --stream-name test --shard-count 1
```

Ketika status aliran adalah `ACTIVE`, aliran sudah siap. Gunakan perintah berikut describe-stream untuk memeriksa status aliran:

```
aws kinesis describe-stream --stream-name test
```

Langkah 2: Buat aturan

Sebagai contoh, buat aturan untuk mengirim peristiwa ke aliran Anda ketika Anda menghentikan instans Amazon EC2.

Untuk membuat tabel

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.

2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola kejadian.
   b. Pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   c. Pilih EC2, Notifikasi Perubahan Status Instans.
   d. Pilih Status khusus, Berjalan.
4. Untuk Target, pilih Tambahkan target, Aliran Kinesis.
5. Untuk Aliran, pilih aliran yang Anda buat.
6. Pilih Membuat peran baru untuk sumber daya khusus ini.
7. Pilih Konfigurasi detail.
8. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
9. Pilih Buat aturan.

Langkah 3: Uji Aturan

Untuk menguji aturan Anda, hentikan satu instans Amazon EC2. Setelah menunggu beberapa menit hingga instans berhenti, periksa metrik CloudWatch Anda untuk memverifikasi bahwa fungsi Anda telah dipanggil.

Untuk menguji aturan Anda dengan pemfilteran stopword instans

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Peluncuran instans. Untuk informasi selengkapnya, lihat Peluncuran Instans Anda di Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
4. Di panel navigasi, pilih Peristiwa, Aturan, lalu pilih nama aturan yang Anda buat, dan kemudian pilih Tampilkan metrik untuk aturan tersebut.
5. (Opsional) Setelah selesai, Anda dapat mengakhiri instans tersebut. Untuk informasi lebih lanjut, lihat Akhiri Instans Anda di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Langkah 4: Verifikasi bahwa Peristiwa Direlai

Anda dapat memperoleh catatan dari aliran untuk memverifikasi bahwa peristiwa telah direlai.

Untuk mendapatkan catatan

1. Gunakan perintah berikut get–shard-iterator untuk mulai membaca dari aliran Kinesis Anda:

```
aws kinesis get-shard-iterator --shard-id shardId-000000000000 --shard-iterator-type
 TRIM_HORIZON --stream-name test
```

   Berikut ini adalah contoh output:

```
{
    "ShardIterator": "AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp
+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd
+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg="
}
```

2. Gunakan perintah berikut get-records untuk mendapatkan catatan. Serpihan iterator adalah hasil yang Anda dapatkan di langkah sebelumnya:

```
aws kinesis get-records --shard-
iterator AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp+KEd9I6AJ9ZG4lNR1EMi
+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd+efGN2aHFdkH1rJl4BL9Wyrk
+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg=
```

Jika perintah berhasil, perintah akan meminta catatan dari aliran Anda untuk shard yang ditentukan. Anda dapat menerim nol atau beberapa catatan. Catatan apa pun yang dikembalikan mungkin tidak mewakili semua catatan di aliran Anda. Jika Anda tidak menerima data yang Anda harapkan, pangill `get-records` secara terus-menerus.

Catatan dalam Kinesis dikodekan sebagai Base64. Namun, dukungan aliran di AWS CLI tidak menyediakan dekode base64. Jika Anda menggunakan decoder base64 untuk memecahkan kode data secara manual, Anda akan melihat bahwa peristiwa tersebut adalah peristiwa yang direlai ke aliran dalam bentuk JSON.

Tutorial: Jalankan Tugas Amazon ECS ketika File Diunggah ke Bucket Amazon S3

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menggunakan CloudWatch Events untuk menjalankan tugas Amazon ECS ketika peristiwa AWS terjadi. Dalam tutorial ini, Anda menyiapkan aturan CloudWatch Events yang menjalankan sebuah tugas Amazon ECS ketika file diunggah ke bucket Amazon S3 tertentu dengan menggunakan operasi Amazon S3 PUT.

Tutorial ini mengasumsikan bahwa Anda telah membuat keterangan tugas di Amazon ECS.

Untuk menjalankan tugas Amazon ECS ketika file diunggah ke bucket S3 menggunakan operasi PUT

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber peristiwa, lakukan hal berikut:
   a. Pilih Pola kejadian.
   b. Pilih Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan.
   c. Untuk Nama Layanan:, pilih Simple Storage Service (S3).
   d. Untuk Jenis Peristiwa, pilih Operasi Tingkat Objek.
   e. Pilih Operasi tertentu, Masukan Object.
   f. Pilih Bucket tertentu berdasarkan nama dan masukkan nama bucket.
4. Untuk Target, lakukan langkah berikut:
   a. Pilih Tambahkan target, Tugas ECS.

b. Untuk Klaster dan Keterangan Tugas, pilih sumber daya yang Anda buat.
c. Untuk Jenis Peluncuran, pilih `FARGATE` atau `EC2`. `FARGATE` hanya muncul di wilayah di mana AWS Fargate didukung.
d. (Opsional) Tentukan nilai untuk Grup Tugas. Jika Jenis Peluncuran adalah `FARGATE`, Anda dapat menentukan Versi Platform. Tentukan hanya bagian numerik dari versi platform, seperti 1.1.0.
e. (Opsional) Tentukan revisi keterangan tugas dan jumlah tugas. Jika Anda tidak menentukan revisi keterangan tugas, maka yang digunakan adalah versi terbaru.
f. Jika keterangan tugas Anda menggunakan mode jaringan awsvpc, Anda harus menentukan subnet dan grup keamanan. Semua subnet dan grup keamanan harus berada di VPC yang sama.

   Jika Anda menentukan lebih dari satu grup keamanan atau subnet, pisahkan dengan koma tetapi jangan gunakan spasi.

   Untuk Subnet, tentukan seluruh nilai `subnet-id` untuk setiap subnet, seperti pada contoh berikut:

   ```
   subnet-123abcd,subnet-789abcd
   ```
g. Pilih apakah Anda akan mengizinkan alamat IP publik ditetapkan secara otomatis.
h. CloudWatch Events dapat membuat IAM role yang diperlukan agar tugas Anda berjalan:

   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan IAM role yang Anda buat sebelumnya, pilih Gunakan peran yang ada. Bagian ini harus menjadi peran yang sudah memiliki izin yang memadai untuk mengkatifkan pembangunan. CloudWatch Events tidak memberikan izin tambahan untuk peran yang Anda pilih.

5. Pilih Konfigurasi detail.
6. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
7. Pilih Buat aturan.

Tutorial: Jadwalkan Pembangunan Otomatis Menggunakan CodeBuild

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Dalam contoh di tutorial ini, Anda menjadwalkan CodeBuild untuk menjalankan pembangunan setiap malam hari kerja pada pukul 20:00 GMT. Anda juga meneruskan konstanta ke CodeBuild yang akan digunakan untuk pembangunan terjadwal ini.

Untuk membuat aturan untuk menjadwalkan pembangunan proyek CodeBuild setiap malam pukul 20:00

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat aturan.
3. Untuk Sumber Peristiwa, lakukan hal berikut:

   a. Pilih Jadwal.

   b. MemiilihEkspresi Crondan tentukan yang berikut sebagai ekspresinya: 0 20? * SEN-JUM *. Untuk informasi selengkapnya tentang ekspresi cron, lihat Ekspresi Jadwal untuk Aturan (p. 33).
4. Untuk Target, pilih Tambahkan target, Proyek CodeBuild.
5. Untuk ARN Proyek, ketik ARN dari proyek pembangunan.
6. Dalam tutorial ini, kami menambahkan langkah opsional untuk meneruskan parameter ke CodeBuild, untuk mengganti default. Hal ini tidak diperlukan ketika Anda menetapkan CodeBuild sebagai target. Untuk meneruskan parameter, pilih Konfigurasi input, Konstanta (teks JSON).

   Di dalam kotak di bawahKonstan (teks JSON), ketik yang berikut ini untuk mengatur penggantian batas waktu ke 30 menit untuk pembangunan terjadwal ini:{“timeoutInMinutesOverride”: 30}

   Untuk informasi selengkapnya tentang parameter yang bisa Anda teruskan, lihat StartBuild. Anda tidak dapat meneruskan parameter `projectName` dalam bidang ini. Alih-alih demikian, Anda menentukan proyek menggunakan ARN di ARN Proyek.
7. CloudWatch Events dapat membuat IAM role yang diperlukan agar proyek pembangunan Anda berjalan:

   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan IAM role yang Anda buat sebelumnya, pilih Gunakan peran yang ada. Bagian ini harus menjadi peran yang sudah memiliki izin yang memadai untuk mengkatifkan pembangunan. CloudWatch Events tidak memberikan izin tambahan untuk peran yang Anda pilih.
8. Pilih Konfigurasi detail
9. Untuk Definisi aturanIT, ketikkan nama dan deskripsi untuk aturan.
10. Pilih Buat aturan.

Tutorial: Mencatat Perubahan Status Instans Amazon EC2

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Dalam contoh di tutorial ini, Anda membuat aturan yang menyebabkan notifikasi perubahan status di Amazon EC2 dicatatkan ke CloudWatch Logs.

Untuk membuat aturan untuk mencatat notifikasi perubahan status Amazon EC2 di CloudWatch Logs

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa lalu Buat aturan.
3. Untuk Sumber Kejadian, lakukan hal berikut:

   a. Pilih Pola kejadian.
   b. Untuk Nama Layanan, pilih EC2 .
   c. Untuk Jenis peristiwa, pilih Notifikasi Perubahan Status Instans EC2.

4. Untuk Targets (Target), pilih Add target (Tambahkan target). Dalam daftar layanan, pilih Grup log CloudWatch.
5. Untuk Grup Log, masukkan nama untuk grup log untuk menerima notifikasi perubahan status.
6. Pilih Konfigurasikan detail.
7. Untuk Definisi aturan, masukkan nama dan deskripsi untuk aturan.
8. Pilih Buat aturan.

Ekspresi Jadwal untuk Aturan

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda juga dapat membuat aturan yang memicu sendiri jadwal yang terotomatisasi di CloudWatch Events menggunakan ekspresi cron atau rate. Semua peristiwa yang dijadwalkan menggunakan zona waktu UTC, dan presisi minimum untuk jadwal adalah satu menit.

CloudWatch Events mendukung ekspresi cron dan ekspresi rate. Ekspresi rate lebih sederhana untuk ditentukan tetapi tidak menawarkan kontrol jadwal detail yang didukung ekspresi cron. Sebagai contoh, dengan ekspresi cron, Anda dapat menentukan aturan yang memicu pada waktu tertentu pada hari tertentu setiap minggu atau bulan. Sebaliknya, ekspresi rate memicu aturan pada tingkat reguler, seperti sekali setiap jam atau sekali setiap hari.

> **Note**
>
> CloudWatch Events tidak memberikan presisi tingkat kedua dalam ekspresi jadwal. Resolusi terbaik untuk penggunaan ekspresi cron adalah satu menit. Karena sifat terdistribusi CloudWatch Events dan layanan target, dapat terjadi penundaan beberapa detik antara waktu ketika aturan yang dijadwalkan dipicu dan waktu ketika layanan target menjalankan sumber daya target. Aturan terjadwal Anda akan dipicu di menit itu tetapi tidak tepat pada detik ke-0.

Format

- Ekspresi Cron (p. 33)
- Ekspresi Rate (p. 36)

Ekspresi Cron

Ekspresi cron memiliki enam bidang yang diperlukan, yang dipisahkan oleh spasi putih.

Sintaks

```
cron(fields)
```

Bidang	Nilai	Wildcard
Menit	0-59	, - * /
Jam	0-23	, - * /
Tanggal	1-31	, - * ? / L W

Bidang	Nilai	Wildcard
Bulan	1-12 atau JAN-DES	, - * /
Hari	1-7 atau MGG-SBT	, - * ? L #
Tahun	1970-2199	, - * /

Wildcard

- Wildcard , (koma) mencakup nilai tambahan. Di field Bulan, JAN, FEB, MAR akan mencakup Januari, Februari, dan Maret.
- Wildcard - (tanda hubung) menentukan rentang. Di field Tanggal, 1-15 akan mencakup tanggal 1 hingga 15 pada bulan yang ditentukan.
- Wildcard * (bintang) mencakup semua nilai di bidang. Di field Jam, * akan mencakup setiap jam. Anda tidak dapat menggunakan * di field Tanggal dan Hari secara bersamaan. Jika Anda menggunakannya di satu bidang, Anda harus menggunakan ? di bidang lain.
- Wildcard / (garis miring) menentukan tambahan. 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 satu atau yang lain. Pada field Hari, Anda dapat memasukkan 7 dan jika Anda tidak memedulikan tanggal 7 jatuh pada hari apa, Anda dapat memasukkan ? di field Hari.
- Wildcard L di bidang Tanggal atau Hari menentukan hari terakhir pada bulan atau minggu.
- Wildcard **`W`** di bidang Tanggal menentukan hari kerja. Di bidang Tanggal, **`3W`** menentukan hari kerja yang paling dekat dengan hari ketiga pada bulan.
- Wildcard # di bidang Hari menentukan instans tertentu dari hari tertentu dalam satu minggu dalam satu bulan. 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 bidang hari. Sebagai contoh, `"3#1,6#3"` tidak valid karena ditafsirkan sebagai dua ekspresi.

Pembatasan

- Anda tidak dapat menentukan bidang Tanggal dan Hari dalam ekspresi cron yang sama. Jika Anda menentukan nilai (atau *) di salah satu field, Anda harus menggunakan ? (tanda tanya) di field lain.
- Ekspresi cron yang mengarah ke tingkat lebih cepat dari 1 menit tidak didukung.

Contoh

Anda dapat menggunakan contoh string cron berikut saat membuat aturan dengan jadwal.

Menit	Jam	Hari dalam sebulan	Bulan	Hari dalam seminggu	Tahun	Arti
0	10	*	*	?	*	Jalankan pada pukul 10:00 pagi (UTC) setiap hari

Menit	Jam	Hari dalam sebulan	Bulan	Hari dalam seminggu	Tahun	Arti
15	12	*	*	?	*	Jalankan pada pukul 12.15 (UTC) setiap hari
0	18	?	*	MON-FRI	*	Jalankan pada pukul 18.00 (UTC) setiap Senin hingga Jumat
0	8	1	*	?	*	Jalankan pada pukul 08.00 (UTC) setiap tanggal 1 pada bulan tersebut
0/15	*	*	*	?	*	Jalankan setiap 15 menit
0/10	*	?	*	MON-FRI	*	Jalankan setiap 10 menit Senin hingga Jumat
0/5	8-17	?	*	MON-FRI	*	Jalankan setiap 5 menit Senin hingga Jumat antara pukul 08.00 dan 17.55 (UTC)

Contoh-contoh berikut menunjukkan cara menggunakan ekspresi Cron dengan perintah AWS CLI put-rule. Contoh pertama ini membuat aturan yang dipicu setiap hari pada pukul 12:00 siang UTC.

```
aws events put-rule --schedule-expression "cron(0 12 * * ? *)" --name MyRule1
```

Contoh pertama ini membuat aturan yang dipicu setiap hari pada pukul 2:05 dan 2:35 siang UTC.

```
aws events put-rule --schedule-expression "cron(5,35 14 * * ? *)" --name MyRule2
```

Contoh berikut membuat aturan yang dipicu pada pukul 10:15 UTC pada hari Jumat terakhir setiap bulan selama tahun 2002 hingga 2005.

```
aws events put-rule --schedule-expression "cron(15 10 ? * 6L 2002-2005)" --name MyRule3
```

Ekspresi Rate

Ekspresi rate dimulai ketika Anda membuat aturan peristiwa terjadwal, dan kemudian aturan berjalan pada jadwal yang ditetapkan.

Ekspresi rate memiliki dua field wajib berikut. Field dipisahkan oleh white space.

Sintaks

```
rate(value unit)
```

nilai

Bilangan positif

unit

Unit waktu. Unit yang berbeda diperlukan untuk nilai 1, seperti `minute`, dan nilai lebih dari 1, seperti `minutes`.

Nilai yang valid: menit | menit-menit | jam | jam-jam | hari | hari-hari

Pembatasan

Jika nilai sama dengan 1, unit harus tunggal. Demikian pula, untuk nilai lebih besar dari 1, unit harus jamak. Misalnya, rate(1 jam) dan rate (5 jam) tidak valid, tetapi rate (1 jam) dan rate(5 jam) valid.

Contoh

Contoh-contoh berikut menunjukkan cara menggunakan ekspresi rate dengan perintah AWS CLI put-rule. Contoh pertama memicu aturan setiap menit, contoh berikutnya memicunya setiap lima menit, contoh ketiga memicunya satu jam sekali, dan contoh terakhir memicunya sekali per hari.

```
aws events put-rule --schedule-expression "rate(1 minute)" --name MyRule2
```

```
aws events put-rule --schedule-expression "rate(5 minutes)" --name MyRule3
```

```
aws events put-rule --schedule-expression "rate(1 hour)" --name MyRule4
```

```
aws events put-rule --schedule-expression "rate(1 day)" --name MyRule5
```

Pola Peristiwa di CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Peristiwa di Amazon CloudWatch Events dilambangkan sebagai objek JSON. Untuk informasi lebih lanjut tentang JSON, lihat RFC 7159. Berikut ini adalah contoh peristiwa:

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/ i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": " i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

Penting untuk mengingat detail berikut tentang suatu peristiwa:

- Semua detail tersebut memiliki field tingkat atas yang sama - yang muncul dalam contoh di atas - yang tidak pernah absen.
- Konten dari bidang tingkat atas detail berbeda bergantung pada layanan yang menghasilkan peristiwa dan apa peristiwanya. Kombinasi dari sumber dan bidang jenis detail berfungsi untuk mengidentifikasi bidang dan nilai yang ditemukan dalam bidang detail. Untuk contoh peristiwa yang dihasilkan oleh layanan AWS, lihat Jenis Peristiwa untuk CloudWatch Events.

Setiap field peristiwa dijelaskan di bawah ini.

versi

Secara default, ini diatur ke 0 (nol) di semua peristiwa.

id

Sebuah nilai unik dihasilkan untuk setiap peristiwa. Hal ini dapat membantu dalam melacak peristiwa saat bergerak dari aturan ke target, dan diproses.

jenis-detail

Mengidentifikasi, dalam kombinasi dengan bidang sumber, bidang dan nilai yang muncul di bidang detail.

Semua peristiwa yang disampaikan oleh CloudTrail memiliki `AWS API Call via CloudTrail` sebagai nilai untuk `detail-type`. Untuk informasi selengkapnya, lihat Peristiwa yang Disampaikan ViaCloudTrail (p. 80).

sumber

Mengidentifikasi layanan yang menghasilkan peristiwa. Semua peristiwa yang bersumber dari dalam AWS dimulai dengan ”AWS.” Peristiwa yang dihasilkan pelanggan dapat memiliki nilai berapa pun di sini, selama tidak dimulai dengan ”AWS.” Kami merekomendasikan penggunaan string nama domain terbalik gaya nama paket Java.

Untuk menemukan nilai yang benar untuk `source` untuk layanan AWS, lihat tabel di Namespaces Layanan AWS. Sebagai contoh, nilai `source` untuk Amazon CloudFront adalah `aws.cloudfront`.

akun

Angka 12 digit yang mengidentifikasi akun AWS.

waktu

Peristiwa timestamp, yang dapat ditentukan oleh layanan yang berasal dari peristiwa. Jika peristiwa mencakup interval waktu, layanan dapat melaporkan waktu mulai, sehingga nilai ini mungkin sebelum waktu peristiwa diterima.

wilayah

Mengidentifikasi wilayah AWS asal peristiwa.

sumber daya

Array JSON berisi ARN yang mengidentifikasi sumber daya yang terlibat dalam peristiwa tersebut. Pencantuman ARN ini tergantung pada kebijakan layanan. Sebagai contoh, perubahan keadaan instans Amazon EC2 termasuk ARN instans Amazon EC2, peristiwa Auto Scaling termasuk ARN untuk kedua instans dan kelompok Auto Scaling, tapi panggilan API dengan AWS CloudTrail tidak mencakup ARN sumber daya.

detail

Sebuah objek JSON, yang berisi kebijakan layanan yang berasal dari peristiwa tersebut. Konten detail dalam contoh di atas sangat sederhana, hanya dua field. AWS Peristiwa panggilan API memiliki objek detail dengan sekitar 50 field yang disarangkan sedalam beberapa tingkat.

Pola Peristiwa

Aturan menggunakan pola kejadian untuk memilih kejadian dan merutekannya ke target. Pola peristiwa cocok atau tidak cocok dengan peristiwa. Pola peristiwa dilambangkan sebagai objek JSON dengan struktur yang mirip dengan struktur peristiwa, misalnya:

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "detail": {
    "state": [ "running" ]
  }
}
```

Penting untuk mengingat detail berikut tentang pencocokan pola peristiwa:

- Agar pola cocok dengan peristiwa, peristiwa harus berisi semua nama field yang tercantum dalam pola peristiwa. Nama field juga harus muncul dalam peristiwa dengan struktur persarangan yang sama.
- Field lain dari peristiwa yang tidak disebutkan dalam pola diabaikan; Secara efektif, terdapat wildcard “*”: “*”: “*”: “*” untuk field yang tidak disebutkan.

- Pencocokan bersifat pasti (karakter per karakter) yang tepat tanpa case-folding atau normalisasi string lainnya.
- Nilai-nilai yang cocok mengikuti aturan JSON: String tertutup dalam tanda kutip, angka, dan kata kunci yang belum dikutip`true`,`false`, dan`null`.
- Angka yang dicocokkan berada di tingkat representasi string. Sebagai contoh, 300, 300,0, dan 3,0e2 tidak dianggap sama.

Ketika Anda menulis pola peristiwa untuk mencocokkan peristiwa, Anda dapat menggunakan API `TestEventPattern` atau perintah CLI `test-event-pattern` untuk memastikan bahwa pola Anda cocok dengan peristiwa yang diinginkan. Untuk informasi selengkapnya, lihat TestEventPattern atau uji-pola-peristiwa.

Pola peristiwa berikut akan cocok dengan peristiwa di bagian atas halaman ini. Pola pertama cocok karena salah satu nilai instans yang ditentukan dalam pola cocok dengan peristiwa (dan pola tidak menentukan field tambahan yang tidak terkandung dalam peristiwa). Pola kedua cocok karena keadaan yang “dihentikan” terkandung dalam peristiwa tersebut.

```
{
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcdefgh"
  ]
}
```

```
{
  "detail": {
    "state": [ "terminated" ]
  }
}
```

Pola peristiwa berikut akan cocok dengan peristiwa di bagian atas halaman ini. Pola pertama tidak cocok karena pola menentukan nilai “tertunda” untuk keadaan, dan nilai ini tidak muncul dalam peristiwa. Pola kedua tidak cocok karena nilai sumber daya yang ditentukan dalam pola tidak muncul dalam peristiwa.

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "detail": {
    "state": [ "pending" ]
  }
}
```

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1::image/ami-12345678" ]
}
```

Pencocokan Nilai Null dan String Kosong dalam Pola Peristiwa

Anda dapat membuat pola peristiwa yang cocok dengan field peristiwa yang memiliki nilai null atau string kosong. Untuk melihat bagaimana proses ini bekerja, pertimbangkan contoh peristiwa berikut:

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [
   ],
  "detail": {
    "eventVersion": "",
    "responseElements": null
   }
}
```

Untuk mencocokkan peristiwa di mana nilai `eventVersion` adalah string kosong, gunakan pola peristiwa berikut ini, yang cocok dengan contoh peristiwa.

```
{
  "detail": {
     "eventVersion": [""]
  }
}
```

Untuk mencocokkan peristiwa di mana nilai `responseElements` adalah null, gunakan pola peristiwa berikut ini, yang cocok dengan contoh peristiwa.

```
{
  "detail": {
     "responseElements": [null]
  }
}
```

Nilai null dan string kosong tidak dapat dipertukarkan dalam pencocokan pola. Sebuah pola yang ditulis untuk mendeteksi string kosong tidak akan menangkap nilai-nilai `null`.

Array dalam Pola CloudWatch Events

Nilai setiap field dalam pola adalah array yang berisi satu atau lebih nilai, dan pola cocok jika salah satu nilai dalam array cocok dengan nilai dalam peristiwa. Jika nilai dalam peristiwa adalah array, pola peristiwa cocok jika persimpangan array pola peristiwa dan array peristiwa tidak kosong.

Misalnya, contoh pola peristiwa termasuk teks berikut:

```
"resources": [
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:111122223333:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:444455556666:instance/i-b188560f",
]
```

Contoh pola peristiwa cocok dengan peristiwa yang mencakup teks berikut ini karena item pertama dalam array pola peristiwa cocok dengan item kedua dalam array peristiwa.

```
"resources": [
   "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/ASGTerminate",
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
]
```

CloudWatchContoh Peristiwa dari Layanan yang Didukung

> **Note**
>
> Amazon EventBridge adalah cara terbaik untuk mengelola peristiwa Anda. CloudWatchPeristiwa danEventBridgeadalah layanan dan API dasar yang sama, tetapiEventBridgemenyediakan lebih banyak fitur. Perubahan yang Anda buat di salah satuCloudWatchatauEventBridgeakan muncul di setiap konsol. Untuk informasi selengkapnya, lihatAmazonEventBridge.

ParameterAWSlayanan dalam daftar berikut memancarkan peristiwa yang dapat dideteksi olehCloudWatchPeristiwa.

Selain itu, Anda juga dapat menggunakanCloudWatchPeristiwa dengan layanan yang tidak menghasilkan peristiwa dan tidak tercantum di halaman ini, dengan menonton peristiwa yang dikirimkan melaluiCloudTrail. Untuk informasi selengkapnya, lihat Peristiwa yang Disampaikan ViaCloudTrail (p. 80).

Jenis Peristiwa

- Peristiwa Amazon Augmented AI (p. 43)
- Peristiwa Application Auto Scaling (p. 43)
- Peristiwa AWS Batch (p. 43)
- AmazonCloudWatchPeristiwa terjadwal (p. 43)
- Peristiwa Amazon Chime (p. 44)
- Peristiwa dariCloudWatch (p. 44)
- CodeBuildPeristiwa (p. 44)
- CodeCommitPeristiwa (p. 44)
- Peristiwa AWS CodeDeploy (p. 44)
- CodePipelinePeristiwa (p. 45)
- Peristiwa AWS Config (p. 46)
- Peristiwa Amazon EBS (p. 46)
- Peristiwa Amazon EC2 Auto Scaling (p. 47)
- Peristiwa Rekomendasi Penyeimbangan Ulang Instans Amazon EC2 (p. 47)
- Peristiwa Interupsi Instans Spot Amazon EC2 (p. 47)
- Peristiwa Perubahan Status Amazon EC2 (p. 47)
- Peristiwa Amazon Elastic Container Registry (p. 47)
- Peristiwa Amazon Elastic Container Service (p. 48)
- AWS ElementalMediaConvertPeristiwa (p. 48)
- AWS ElementalMediaPackagePeristiwa (p. 48)
- AWS ElementalMediaStorePeristiwa (p. 48)
- Peristiwa Amazon EMR (p. 48)
- AmazonGameLiftPeristiwa (p. 50)
- Peristiwa AWS Glue (p. 57)

- Peristiwa AWS Ground Station (p. 62)
- Amazon GuardDuty Events (p. 62)
- Peristiwa AWS Health (p. 62)
- Peristiwa AWS KMS (p. 64)
- Peristiwa Amazon Macie (p. 65)
- Peristiwa Masuk AWS Management Console (p. 65)
- Peristiwa Tumpukan AWS OpsWorks (p. 66)
- SageMakerPeristiwa (p. 68)
- Peristiwa AWS Security Hub (p. 68)
- Peristiwa AWS Server Migration Service (p. 68)
- Peristiwa AWS Systems Manager (p. 69)
- Peristiwa AWS Step Functions (p. 78)
- Tandai Perubahan Peristiwa di Sumber Daya AWS (p. 78)
- Peristiwa AWS Trusted Advisor (p. 78)
- WorkSpacesPeristiwa (p. 80)
- Peristiwa yang Disampaikan ViaCloudTrail (p. 80)

Peristiwa Amazon Augmented AI

Untuk contoh peristiwa yang dihasilkan oleh Amazon Augmented AI, lihat Gunakan Peristiwa di Amazon Augmented AI.

Peristiwa Application Auto Scaling

Untuk contoh peristiwa yang dihasilkan oleh Application Auto Scaling, lihatPeristiwa Application Auto Scaling danEventBridge.

Peristiwa AWS Batch

Untuk contoh peristiwa yang dihasilkan oleh layanan AWS Batch, lihat AWS Batch Peristiwa.

AmazonCloudWatchPeristiwa terjadwal

Berikut ini adalah contoh peristiwa terjadwal:

```
{
  "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  "account": "123456789012",
  "time": "2019-10-08T16:53:06Z",
  "region": "us-east-1",
  "resources": [ "arn:aws:events:us-east-1:123456789012:rule/MyScheduledRule" ],
  "detail": {}
}
```

Peristiwa Amazon Chime

Untuk contoh peristiwa yang dihasilkan oleh Amazon Chime, lihatMengotomatisasi Amazon Chime denganEventBridge.

Peristiwa dariCloudWatch

Untuk peristiwa sampel dariCloudWatch, lihatPeristiwa Alarm danEventBridgediAWS CodeBuildPanduan Pengguna.

CodeBuildPeristiwa

UntukCodeBuildcontoh peristiwa, lihatMembangun Referensi Format Input PemberitahuandiAWS CodeBuildPanduan Pengguna.

CodeCommitPeristiwa

UntukCodeCommitcontoh peristiwa, lihatPemantauanCodeCommitPeristiwa diEventBridgedanCloudWatchPeristiwadiAWS CodeCommitPanduan Pengguna.

Peristiwa AWS CodeDeploy

Berikut ini adalah contoh peristiwa untuk CodeDeploy. Untuk informasi selengkapnya, lihatMemantau Deployment denganCloudWatchPeristiwadiAWS CodeDeployPanduan Pengguna.

CodeDeployNotifikasi Perubahan Status Deployment

Ada perubahan dalam status deployment.

```
{
  "account": "123456789012",
  "region": "us-east-1",
  "detail-type": "CodeDeploy Deployment State-change Notification",
  "source": "aws.codedeploy",
  "version": "0",
  "time": "2016-06-30T22:06:31Z",
  "id": "c071bfbf-83c4-49ca-a6ff-3df053957145",
  "resources": [
    "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication",
    "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/
myDeploymentGroup"
  ],
  "detail": {
    "instanceGroupId": "9fd2fbef-2157-40d8-91e7-6845af69e2d2",
    "region": "us-east-1",
    "application": "myApplication",
    "deploymentId": "d-123456789",
    "state": "SUCCESS",
    "deploymentGroup": "myDeploymentGroup"
  }
}
```

CodeDeployNotifikasi Perubahan Status Instans

Ada perubahan dalam status instans yang dimiliki oleh kelompok deployment.

```
{
  "account": "123456789012",
  "region": "us-east-1",
  "detail-type": "CodeDeploy Instance State-change Notification",
  "source": "aws.codedeploy",
  "version": "0",
  "time": "2016-06-30T23:18:50Z",
  "id": "fb1d3015-c091-4bf9-95e2-d98521ab2ecb",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa",
    "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/
myDeploymentGroup",
    "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication"
  ],
  "detail": {
    "instanceId": "i-0000000aaaaaaaaaa",
    "region": "us-east-1",
    "state": "SUCCESS",
    "application": "myApplication",
    "deploymentId": "d-123456789",
    "instanceGroupId": "8cd3bfa8-9e72-4cbe-a1e5-da4efc7efd49",
    "deploymentGroup": "myDeploymentGroup"
  }
}
```

CodePipelinePeristiwa

Berikut ini adalah contoh peristiwa untukCodePipeline.

Perubahan Status Eksekusi Alur

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Pipeline Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": "1",
    "state": "STARTED",
    "execution-id": "01234567-0123-0123-0123-012345678901"
  }
}
```

Perubahan Status Eksekusi Tahap

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Stage Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
```

```
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": "1",
    "execution-id": "01234567-0123-0123-0123-012345678901",
    "stage": "Prod",
    "state": "STARTED"
  }
}
```

Perubahan Status Eksekusi Tindakan

Dalam contoh ini, ada dua bidang `region`. Yang di atas adalah nama Wilayah AWS tempat tindakan aksi dalam alur target dijalankan. Dalam contoh ini, ini adalah `us-east-1`. `region` dalam bagian `detail` adalah Wilayah AWS tempat peristiwa dibuat. Ini sama dengan Wilayah tempat alur dibuat. Dalam contoh ini, ini adalah `us-west-2`.

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Action Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": 1,
    "execution-id": "01234567-0123-0123-0123-012345678901",
    "stage": "Prod",
    "action": "myAction",
    "state": "STARTED",
    "region":"us-west-2",
    "type": {
      "owner": "AWS",
      "category": "Deploy",
      "provider": "CodeDeploy",
      "version": 1
    }
  }
}
```

Peristiwa AWS Config

Untuk informasi tentangAWS Configperistiwa, lihatPemantauanAWS Configdengan AmazonCloudWatchPeristiwadiAWS ConfigPanduan Pengembang.

Peristiwa Amazon EBS

Untuk informasi tentang peristiwa Amazon EBS, lihatAmazonCloudWatchEvent untuk Amazon EBSdiPanduan Pengguna Amazon EC2 untuk Instans Linux.

Peristiwa Amazon EC2 Auto Scaling

Untuk informasi tentang peristiwa Auto Scaling, lihatMendapatkanCloudWatchPeristiwa Saat Scaling Group Scaling Otomatis AndadiPanduan Pengguna Amazon EC2 Auto Scaling.

Peristiwa Rekomendasi Penyeimbangan Ulang Instans Amazon EC2

Untuk informasi tentang peristiwa untuk rekomendasi penyeimbangan ulang instans EC2, lihat Memantau tanda rekomendasi penyeimbangan ulang di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Peristiwa Interupsi Instans Spot Amazon EC2

Untuk informasi tentang peristiwa interupsi Instans Spot, lihat pemberitahuan interupsi Instans Spot dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Peristiwa Perubahan Status Amazon EC2

Berikut ini adalah contoh kejadian untuk instans Amazon EC2 apabila status instans berubah.

Notifikasi Perubahan Status Instans EC2

Contoh ini adalah untuk sebuah instans di status `pending`. Nilai lain yang mungkin untuk `state` termasuk `running`, `shutting-down`, `stopped`, `stopping`, dan `terminated`.

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2019-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
   ],
   "detail":{
      "instance-id":"i-abcd1111",
      "state":"pending"
   }
}
```

Peristiwa Amazon Elastic Container Registry

Amazon ECR mengirimkan peristiwa tindakan gambar keEventBridge. Peristiwa dikirim ketika citra dikirim, dipindai, atau dihapus.

Untuk contoh peristiwa Amazon ECS, lihat Peristiwa Amazon ECR di Panduan Pengguna Registri Kontainer Elastis.

Peristiwa Amazon Elastic Container Service

Amazon ECS mengirimkan dua jenis peristiwaEventBridge: peristiwa instans kontainer dan peristiwa tugas. Peristiwa instans kontainer hanya dikirim jika Anda menggunakan jenis peluncuran EC2 untuk tugas Anda. Untuk tugas yang menggunakan jenis peluncuran Fargate, Anda hanya menerima peristiwa status tugas. Amazon ECS melacak status instans kontainer dan tugas. Jika sumber daya ini berubah, sebuah peristiwa terpicu. Peristiwa ini diklasifikasikan sebagai peristiwa perubahan status instans kontainer or peristiwa perubahan status tugas.

Untuk contoh peristiwa Amazon ECS, lihat Peristiwa Amazon ECS di Panduan Developer Amazon Elastic Container Service.

AWS ElementalMediaConvertPeristiwa

UntukMediaConvertcontoh peristiwa, lihatMenggunakanCloudWatchAcara untuk Memantau AWS ElementalMediaConvertTugasdiAWS ElementalMediaConvertPanduan Pengguna.

AWS ElementalMediaPackagePeristiwa

UntukMediaPackagecontoh peristiwa, lihatPemantauan Elemen AWSMediaPackagedengan AmazonCloudWatchPeristiwadiAWS ElementalMediaPackagePanduan Pengguna.

AWS ElementalMediaStorePeristiwa

UntukMediaStorecontoh peristiwa, lihatMengotomatisasi Elemen AWSMediaStorebersamaCloudWatchPeristiwadiAWS ElementalMediaStorePanduan Pengguna.

Peristiwa Amazon EMR

Peristiwa yang dilaporkan oleh Amazon EMR`aws.emr`sebagai nilai untuk`Source`, sementara peristiwa Amazon EMR API dilaporkan melaluiCloudTrailmemilikiaws.elasticmapreducesebagai nilai untuk`Source`.

Berikut ini adalah contoh peristiwa yang dilaporkan oleh Amazon EMR.

Perubahan Status Kebijakan Auto Scaling Amazon EMR

```
{
   "version":"0",
   "id":"2f8147ab-8c48-47c6-b0b6-3ee23ec8d300",
   "detail-type":"EMR Auto Scaling Policy State Change",
   "source":"aws.emr",
   "account":"123456789012",
   "time":"2016-12-16T20:42:44Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "resourceId":"ig-X2LBMHTGPCBU",
      "clusterId":"j-1YONHTCP3YZKC",
      "state":"PENDING",
      "message":"AutoScaling policy modified by user request",
```

```
      "scalingResourceType":"INSTANCE_GROUP"
   }
}
```

Perubahan Status Klaster Amazon EMR - Mulai

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:43:05Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "STARTING",
    "message": "Amazon EMR cluster j-123456789ABCD (Development Cluster) was requested at
 2016-12-16 20:42 UTC and  is being created."
  }
}
```

Perubahan Status Klaster Amazon EMR — Dihentikan

```
{
  "version": "0",
  "id": "1234abb0-f87e-1234-b7b6-000000123456",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T21:00:23Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"USER_REQUEST\",\"message\":\"Terminated by user
 request\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "TERMINATED",
    "message": "Amazon EMR Cluster jj-123456789ABCD (Development Cluster) has terminated at
 2016-12-16 21:00 UTC with a reason of USER_REQUEST."
  }
}
```

Perubahan Status Grup Instans Amazon EMR

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Instance Group State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:57:47Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "market": "ON_DEMAND",
```

```
    "severity": "INFO",
    "requestedInstanceCount": "2",
    "instanceType": "m3.xlarge",
    "instanceGroupType": "CORE",
    "instanceGroupId": "ig-ABCDEFGHIJKL",
    "clusterId": "j-123456789ABCD",
    "runningInstanceCount": "2",
    "state": "RUNNING",
    "message": "The resizing operation for instance group ig-ABCDEFGHIJKL in Amazon EMR
 cluster j-123456789ABCD (Development Cluster) is complete. It now has an instance count of
 2. The resize started at 2016-12-16 20:57 UTC and took 0 minutes to complete."
  }
}
```

Perubahan Status Langkah Amazon EMR

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Step Status Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:53:09Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "ERROR",
    "actionOnFailure": "CONTINUE",
    "stepId": "s-ZYXWVUTSRQPON",
    "name": "CustomJAR",
    "clusterId": "j-123456789ABCD",
    "state": "FAILED",
    "message": "Step s-ZYXWVUTSRQPON (CustomJAR) in Amazon EMR cluster j-123456789ABCD
 (Development Cluster) failed at 2016-12-16 20:53 UTC."
  }
}
```

AmazonGameLiftPeristiwa

Berikut ini adalah contoh AmazonGameLiftperistiwa. Untuk informasi selengkapnya, lihatFlexMatchReferensi PeristiwadiAmazonGameLiftPanduan Pengembang.

Pencocokan Pencarian

```
{
  "version": "0",
  "id": "cc3d3ebe-1d90-48f8-b268-c96655b8f013",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T21:15:36.421Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T21:15:35.676Z",
```

```
        "players": [
          {
            "playerId": "player-1"
          }
        ]
      }
    ],
    "estimatedWaitMillis": "NOT_AVAILABLE",
    "type": "MatchmakingSearching",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1"
        }
      ]
    }
  }
}
```

Pencocokan Potensi Dibuat

```
{
  "version": "0",
  "id": "fce8633f-aea3-45bc-aeba-99d639cad2d4",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T21:17:41.178Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T21:15:35.676Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-08T21:17:40.657Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue"
          }
        ]
      }
    ],
    "acceptanceTimeout": 600,
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 3,
```

```
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 3,
        "failedCount": 0
      }
    ],
    "acceptanceRequired": true,
    "type": "PotentialMatchCreated",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "team": "blue"
        }
      ]
    },
    "matchId": "3faf26ac-f06e-43e5-8d86-08feff26f692"
  }
}
```

Menerima Pencocokan

```
{
  "version": "0",
  "id": "b3f76d66-c8e5-416a-aa4c-aa1278153edc",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:04:42.660Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T20:01:35.305Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T20:04:16.637Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue",
            "accepted": false
          }
```

```
        ]
      }
    ],
    "type": "AcceptMatch",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "team": "blue",
          "accepted": false
        }
      ]
    },
    "matchId": "848b5f1f-0460-488e-8631-2960934d13e5"
  }
}
```

Menerima Pencocokan Selesai

```
{
  "version": "0",
  "id": "b1990d3d-f737-4d6c-b150-af5ace8c35d3",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T20:43:14.621Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T20:30:40.972Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-08T20:33:14.111Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue"
          }
        ]
      }
    ],
    "acceptance": "TimedOut",
    "type": "AcceptMatchCompleted",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
```

```
        {
          "playerId": "player-2",
          "team": "blue"
        }
      ]
    },
    "matchId": "a0d9bd24-4695-4f12-876f-ea6386dd6dce"
  }
}
```

Pencocokan Berhasil

```
{
  "version": "0",
  "id": "5ccb6523-0566-412d-b63c-1569e00d023d",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T19:59:09.159Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T19:58:59.277Z",
        "players": [
          {
            "playerId": "player-1",
            "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T19:59:08.663Z",
        "players": [
          {
            "playerId": "player-2",
            "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
            "team": "blue"
          }
        ]
      }
    ],
    "type": "MatchmakingSucceeded",
    "gameSessionInfo": {
      "gameSessionArn": "arn:aws:gamelift:us-west-2:123456789012:gamesession/836cf48d-
bcb0-4a2c-bec1-9c456541352a",
      "ipAddress": "192.168.1.1",
      "port": 10777,
      "players": [
        {
          "playerId": "player-1",
          "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
          "team": "blue"
        }
```

```
      ]
    },
    "matchId": "c0ec1a54-7fec-4b55-8583-76d67adb7754"
  }
}
```

Pencocokan Waktu Habis

```
{
  "version": "0",
  "id": "fe528a7d-46ad-4bdc-96cb-b094b5f6bf56",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:11:35.598Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "reason": "TimedOut",
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T20:01:35.305Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      }
    ],
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 3,
        "failedCount": 0
      }
    ],
    "type": "MatchmakingTimedOut",
    "message": "Removed from matchmaking due to timing out.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    }
```

```
  }
}
```

Perjodohan Dibatalkan

```
{
  "version": "0",
  "id": "8d6f84da-5e15-4741-8d5c-5ac99091c27f",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:00:07.843Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "reason": "Cancelled",
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T19:59:26.118Z",
        "players": [
          {
            "playerId": "player-1"
          }
        ]
      }
    ],
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 0,
        "failedCount": 0
      }
    ],
    "type": "MatchmakingCancelled",
    "message": "Cancelled by request.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1"
        }
      ]
    }
  }
}
```

Perjodohan Gagal

```
{
  "version": "0",
  "id": "025b55a4-41ac-4cf4-89d1-f2b3c6fd8f9d",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-16T18:41:09.970Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-16T18:41:02.631Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      }
    ],
    "customEventData": "foo",
    "type": "MatchmakingFailed",
    "reason": "UNEXPECTED_ERROR",
    "message": "An unexpected error was encountered during match placing.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    },
    "matchId": "3ea83c13-218b-43a3-936e-135cc570cba7"
  }
}
```

Peristiwa AWS Glue

Berikut ini adalah format untuk peristiwa AWS Glue.

Jalankan Tugas Berhasil

```
{
    "version":"0",
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-09-07T18:57:21Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"INFO",
        "state":"SUCCEEDED",
        "jobRunId":"jr_abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789",
        "message":"Job run succeeded"
```

```
    }
}
```

Jalankan Tugas Gagal

```
{
    "version":"0",
    "id":"abcdef01-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-09-07T06:02:03Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"ERROR",
        "state":"FAILED",
        "jobRunId":"jr_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef",
        "message":"JobName:MyJob and
 JobRunId:jr_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef failed to
 execute with exception Role arn:aws:iam::123456789012:role/Glue_Role should be given
 assume role permissions for Glue Service."
    }
}
```

Waktu habis

```
{
    "version":"0",
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-11-20T20:22:06Z",
    "region":"us-east-1",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"WARN",
        "state":"TIMEOUT",
        "jobRunId":"jr_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
        "message":"Job run timed out"
    }
}
```

Jalankan Tugas Berhenti

```
{
   "version":"0",
   "id":"abcdef00-1234-5678-9abc-def012345678",
   "detail-type":"Glue Job State Change",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2017-11-20T20:22:06Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
       "jobName":"MyJob",
       "severity":"INFO",
       "state":"STOPPED",
       "jobRunId":"jr_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
```

```
        "message":"Job run stopped"
    }
}
```

Crawler Mulai

```
{
   "version":"0",
   "id":"05efe8a2-c309-6884-a41b-3508bcdc9695",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"561226563745",
   "time":"2017-11-11T01:09:46Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "accountId":"561226563745",
      "crawlerName":"S3toS3AcceptanceTestCrawlera470bd94-9e00-4518-8942-e80c8431c322",
      "startTime":"2017-11-11T01:09:46Z",
      "state":"Started",
      "message":"Crawler Started"
   }
}
```

Crawler Berhasil

```
{
   "version":"0",
   "id":"3d675db5-59b9-6388-b8e8-e0a9b6d567a9",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"561226563745",
   "time":"2017-11-11T01:25:00Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "tablesCreated":"0",
      "warningMessage":"N/A",
      "partitionsUpdated":"0",
      "tablesUpdated":"0",
      "message":"Crawler Succeeded",
      "partitionsDeleted":"0",
      "accountId":"561226563745",
      "runningTime (sec)":"7",
      "tablesDeleted":"0",
      "crawlerName":"SchedulerTestCrawler51fb3a8b-1015-49f0-a969-ca126680b94b",
      "completionDate":"2017-11-11T01:25:00Z",
      "state":"Succeeded",
      "partitionsCreated":"0",
      "cloudWatchLogLink":"https://console.aws.amazon.com/
cloudwatch/home?region=us-east-1#logEventViewer:group=/aws-glue/
crawlers;stream=SchedulerTestCrawler51fb3a8b-1015-49f0-a969-ca126680b94b"
   }
}
```

Crawler Gagal

```
{
```

```
   "version":"0",
   "id":"f7965b59-470f-2e06-bb89-a8cebaabefac",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"782104008917",
   "time":"2017-10-20T05:10:08Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "crawlerName":"test-crawler-notification",
      "errorMessage":"Internal Service Exception",
      "accountId":"1234",
      "cloudWatchLogLink":"https://console.aws.amazon.com/cloudwatch/home?region=us-
east-1#logEventViewer:group=/aws-glue/crawlers;stream=test-crawler-notification",
      "state":"Failed",
      "message":"Crawler Failed"
   }
}
```

Jalankan Tugas dalam Status Mulai

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"STARTING",
          "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in STARTING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

Jalankan Tugas dalam Status Berjalan

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"RUNNING",
```

```
    "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in RUNNING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

Jalankan Tugas dalam Status Berhenti

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"STOPPING",
    "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in STOPPING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

AWS GluePerubahan Status Tabel Katalog Data

```
{
    "version": "0",
    "id": "2617428d-715f-edef-70b8-d210da0317a0",
    "detail-type": "Glue Data Catalog Table State Change",
    "source": "aws.glue",
    "account": "123456789012",
    "time": "2019-01-16T18:16:01Z",
    "region": "eu-west-1",
    "resources": [
        "arn:aws:glue:eu-west-1:123456789012:table/d1/t1"
    ],
    "detail": {
        "databaseName": "d1",
        "changedPartitions": [
            "[C.pdf, dir3]",
            "[D.doc, dir4]"
        ],
        "typeOfChange": "BatchCreatePartition",
        "tableName": "t1"
    }
}
```

AWS GluePerubahan Status Basis Data Katalog Data

Pada contoh berikut, `typeofChange` adalah `CreateTable`. Kemungkinan nilai lain untuk field ini adalah `CreateDatabase` and `UpdateTable`.

```
{
    "version": "0",
    "id": "60e7ddc2-a588-5328-220a-21c060f6c3f4",
```

```
    "detail-type": "Glue Data Catalog Database State Change",
    "source": "aws.glue",
    "account": "123456789012",
    "time": "2019-01-16T18:08:48Z",
    "region": "eu-west-1",
    "resources": [
        "arn:aws:glue:eu-west-1:123456789012:table/d1/t1"
    ],
    "detail": {
        "databaseName": "d1",
        "typeOfChange": "CreateTable",
        "changedTables": [
            "t1"
        ]
    }
}
```

Peristiwa AWS Ground Station

Untuk informasi tentang contohAWS Ground Stationperistiwa, lihatMengotomatisasiAWS Ground StationbersamaCloudWatchPeristiwadiAWS Ground StationPanduan Pengguna.

Amazon GuardDuty Events

Untuk informasi tentang contoh AmazonGuardDutyperistiwa, lihatPemantauan AmazonGuardDutydengan AmazonCloudWatchPeristiwadiAmazonGuardDutyPanduan Pengguna.

Peristiwa AWS Health

Berikut ini adalah format untuk peristiwa Personal Health Dashboard AWS (AWS Health). Untuk informasi selengkapnya, lihatMengelolaAWS HealthPeristiwa dengan AmazonCloudWatchPeristiwadiAWS HealthPanduan Pengguna.

AWS HealthFormat Peristiwa

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "region",
  "resources": [],
  "detail": {
    "eventArn": "arn:aws:health:region::event/id",
    "service": "service",
    "eventTypeCode": "AWS_service_code",
    "eventTypeCategory": "category",
    "startTime": "Sun, 05 Jun 2016 05:01:10 GMT",
    "endTime": "Sun, 05 Jun 2016 05:30:57 GMT",
    "eventDescription": [{
      "language": "lang-code",
      "latestDescription": "description"
    }]
```

```
    ...
  }
}
```

eventTypeCategory

Kode kategori peristiwa. Nilai yang mungkin adalah `issue`, `accountNotification`, dan `scheduledChange`.

eventTypeCode

Pengidentifikasi unik untuk jenis peristiwa tersebut. Contohnya termasuk `AWS_EC2_INSTANCE_NETWORK_MAINTENANCE_SCHEDULED` dan `AWS_EC2_INSTANCE_REBOOT_MAINTENANCE_SCHEDULED`. Peristiwa yang mencakup `MAINTENANCE_SCHEDULED` biasanya dibuat sekitar dua minggu sebelum `startTime`.

id

Pengidentifikasi unik untuk peristiwa tersebut.

layanan

Layanan AWS yang dipengaruhi oleh peristiwa tersebut. Contohnya: `EC2`, `S3`, `REDSHIFT`, atau `RDS`.

Masalah API Elastic Load Balancing

```
{
  "version": "0",
  "id": "121345678-1234-1234-1234-123456789012",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "ap-southeast-2",
  "resources": [],
  "detail": {
    "eventArn": "arn:aws:health:ap-southeast-2::event/
AWS_ELASTICLOADBALANCING_API_ISSUE_90353408594353980",
    "service": "ELASTICLOADBALANCING",
    "eventTypeCode": "AWS_ELASTICLOADBALANCING_API_ISSUE",
    "eventTypeCategory": "issue",
    "startTime": "Sat, 11 Jun 2016 05:01:10 GMT",
    "endTime": "Sat, 11 Jun 2016 05:30:57 GMT",
    "eventDescription": [{
      "language": "en_US",
      "latestDescription": "A description of the event will be provided here"
  }
}
```

Performa Drive Penyimpanan Instans Amazon EC2 Menurun

```
{
  "version": "0",
  "id": "121345678-1234-1234-1234-123456789012",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "us-west-2",
  "resources": [
    "i-abcd1111"
  ],
  "detail": {
```

```
    "eventArn": "arn:aws:health:us-west-2::event/
AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED_90353408594353980",
    "service": "EC2",
    "eventTypeCode": "AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED",
    "eventTypeCategory": "issue",
    "startTime": "Sat, 05 Jun 2016 15:10:09 GMT",
    "eventDescription": [{
      "language": "en_US",
      "latestDescription": "A description of the event will be provided here"
    }],
    "affectedEntities": [{
      "entityValue": "i-abcd1111",
      "tags": {
        "stage": "prod",
        "app": "my-app"
  }
}
```

Peristiwa AWS KMS

Berikut ini adalah contoh peristiwa AWS Key Management Service (AWS KMS). Untuk informasi selengkapnya, lihat Peristiwa AWS KMS dalam Panduan Developer AWS Key Management Service.

Rotasi CMK KMS

AWS KMS secara otomatis memutar bahan kunci CMK.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-25T21:05:33Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

KMS Mengimpor Tanggal Kedaluwarsa Bahan Kunci

AWS KMS menghapus bahan kunci CMK yang kedaluwarsa.

```
{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-22T20:12:19Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
```

```
}
```

Penghapusan CMK KMS

AWS KMS menyelesaikan penghapusan CMK yang terjadwal.

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-19T03:23:45Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Peristiwa Amazon Macie

Untuk contoh peristiwa yang dihasilkan oleh Amazon Macie, lihat Skema peristiwa untuk temuan Amazon Macie.

Peristiwa Masuk AWS Management Console

AWS Management Consoleperistiwa masuk dapat dideteksi olehCloudWatchPeristiwa hanya di US East (N. Virginia).

Berikut ini adalah contoh peristiwa masuk konsol:

```
{
  "id": "6f87d04b-9f74-4f04-a780-7acf4b0a9b38",
  "detail-type": "AWS Console Sign In via CloudTrail",
  "source": "aws.signin",
  "account": "123456789012",
  "time": "2016-01-05T18:21:27Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "Root",
            "principalId": "123456789012",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012"
            },
      "eventTime": "2016-01-05T18:21:27Z",
      "eventSource": "signin.amazonaws.com",
      "eventName": "ConsoleLogin",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "0.0.0.0",
      "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36",
      "requestParameters": null,
```

```
      "responseElements": {
            "ConsoleLogin": "Success"
            },
      "additionalEventData": {
            "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
            "MobileVersion": "No",
            "MFAUsed": "No" },
      "eventID": "324731c0-64b3-4421-b552-dfc3c27df4f6",
      "eventType": "AwsConsoleSignIn"
      }
}
```

Peristiwa Tumpukan AWS OpsWorks

Berikut ini adalah contoh peristiwa Tumpukan AWS OpsWorks.

AWS OpsWorksPerubahan status instans tumpukan

Menunjukkan perubahan di status instans Tumpukan AWS OpsWorks. Berikut ini adalah keadaan instans.

- `booting`
- `connection_lost`
- `online`
- `pending`
- `rebooting`
- `requested`
- `running_setup`
- `setup_failed`
- `shutting_down`
- `start_failed`
- `stopping`
- `stop_failed`
- `stopped`
- `terminating`
- `terminated`

```
{
  "version": "0",
  "id": "dc5fa8df-48f1-2108-b1b9-1fe5ebcf2296",
  "detail-type": "OpsWorks Instance State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-25T11:12:23Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50z3e4z500z"
  ],
  "detail": {
    "initiated_by": "user",
    "hostname": "testing1",
    "stack-id": "acd3df16-e859-4598-8414-377b12a902da",
    "layer-ids": [
      "d1a0cb7f-c7e9-4a63-811c-976f0267b2c8"
    ],
```

```
    "instance-id": "a648d98f-fdd8-4323-952a-a50z3e4z500z",
    "ec2-instance-id": "i-08b1c2b67aa292276",
    "status": "requested"
  }
}
```

Bidang `initiated_by` hanya diisi ketika instans dalam status `requested`, `terminating`, atau `stopping`. Field `initiated_by` dapat berisi salah satu dari nilai berikut.

- `user` - Seorang pengguna meminta perubahan keadaan contoh dengan menggunakan baik API atau AWS Management Console.
- `auto-scaling` - Fitur penskalaan otomatis AWS OpsWorks memulai perubahan status instans.
- `auto-healing` - Fitur penyembuhan otomatis AWS OpsWorks memulai perubahan status instans.

AWS OpsWorksPerubahan status perintah tumpukan

Sebuah perubahan terjadi di status perintah Tumpukan AWS OpsWorks. Berikut ini adalah status perintah.

- `expired` - Sebuah perintah waktu habis.
- `failed` - Sebuah kegagalan perintah umum terjadi.
- `skipped` - Sebuah perintah dilewati karena instans memiliki status yang berbeda di Tumpukan AWS OpsWorks dibandingkan di Amazon EC2.
- `successful` - Sebuah perintah berhasil.
- `superseded` - Perintah dilewati karena akan menerapkan perubahan konfigurasi yang telah diterapkan.

```
{
  "version": "0",
  "id": "96c778b6-a40e-c8c1-aafc-c9852a3a7b52",
  "detail-type": "OpsWorks Command State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-26T08:54:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50a3e4e500f"
  ],
  "detail": {
    "command-id": "acc9f4f3-a3ec-4fab-b70f-c7d04e71e3ec",
    "instance-id": "a648d98f-fdd8-4323-952a-a50a3e4e500f",
    "type": "setup",
    "status": "successful"
  }
}
```

AWS OpsWorksPerubahan status deployment Tumpukan

Sebuah perubahan terjadi di status deployment Tumpukan AWS OpsWorks. Berikut ini adalah status deployment.

- `running`
- `successful`
- `failed`

```
{
  "version": "0",
  "id": "b8230afa-60c7-f43f-b632-841c1cfb22ff",
```

```
  "detail-type": "OpsWorks Deployment State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-25T11:15:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50a3e4e500f"
  ],
  "detail": {
    "duration": 16,
    "stack-id": "acd3df16-e859-4598-8414-377b12a902da",
    "instance-ids": [
      "a648d98f-fdd8-4323-952a-a50a3e4e500f"
    ],
    "deployment-id": "606419dc-418e-489c-8531-bff9770fc346",
    "command": "configure",
    "status": "successful"
  }
}
```

Field `duration` hanya diisi ketika deployment selesai, dan menunjukkan waktu dalam detik.

AWS OpsWorksPemberitahuan tumpukan

Kesalahan layanan Tumpukan AWS OpsWorks dimunculkan.

```
{
  "version": "0",
  "id": "f99faa6f-0e27-e398-95bb-8f190806d275",
  "detail-type": "OpsWorks Alert",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-20T16:51:29Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "stack-id": "2f48f2be-ac7d-4dd5-80bb-88375f94db7b",
    "instance-id": "986efb74-69e8-4c6d-878e-5b77c054cbb0",
    "type": "InstanceStop",
    "message": "The shutdown of the instance timed out. Please try stopping it again."
  }
}
```

SageMakerPeristiwa

Untuk informasi tentang contohSageMakerperistiwa, lihatMengotomatisasiSageMakerdengan AmazonEventBridgediSageMakerPanduan Pengembang

Peristiwa AWS Security Hub

Untuk informasi tentang contoh peristiwa Security Hub, lihatPemantauanAWS Security Hubdengan AmazonCloudWatchPeristiwadiAWS Security HubPanduan Pengguna.

Peristiwa AWS Server Migration Service

Berikut ini adalah contoh peristiwa untuk AWS Server Migration Service.

Notifikasi tugas replikasi dihapus

```
{
    "version": "0",
    "id": "5630992d-92cd-439f-f2a8-92c8212aee24",
    "detail-type": "Server Migration Job State Change",
    "source": "aws.sms",
    "account": "123456789012",
    "time": "2018-02-07T22:30:11Z",
    "region": "us-west-1",
    "resources": [
        "arn:aws:sms:us-west-1:123456789012:sms-job-21a64348"
    ],
    "detail": {
        "state": "Deleted",
        "replication-run-id": "N/A",
        "replication-job-id": "sms-job-21a64348",
        "version": "1.0"
    }
}
```

Notifikasi tugas replikasi selesai

```
{
    "version": "0",
    "id": "3f9c59cc-f941-522a-be6d-f08e44ff1715",
    "detail-type": "Server Migration Job State Change",
    "source": "aws.sms",
    "account": "123456789012",
    "time": "2018-02-07T22:54:00Z",
    "region": "us-west-1",
    "resources": [
        "arn:aws:sms:us-west-1:123456789012:sms-job-2ea64347",
        "arn:aws:sms:us-west-1:123456789012:sms-job-2ea64347/sms-run-e1a64388"
    ],
    "detail": {
        "state": "Completed",
        "replication-run-id": "sms-run-e1a64388",
        "replication-job-id": "sms-job-2ea64347",
        "ami-id": "ami-746d6314",
        "version": "1.0"
    }
}
```

Peristiwa AWS Systems Manager

Berikut ini adalah contoh peristiwa untuk AWS Systems Manager. Untuk informasi selengkapnya, lihatPemantauan peristiwa Systems Manager dengan AmazonEventBridgediAWS Systems ManagerPanduan Pengguna.

Jenis peristiwa Systems Manager

- Peristiwa Otomatisasi AWS Systems Manager (p. 70)
- Peristiwa Kalender Perubahan AWS Systems Manager (p. 70)
- Peristiwa Kepatuhan AWS Systems Manager (p. 71)
- Peristiwa Windows Maintenance AWS Systems Manager (p. 73)
- Peristiwa Menyimpan Parameter AWS Systems Manager (p. 75)
- Peristiwa Run Command AWS Systems Manager (p. 76)

- Peristiwa State Manager AWS Systems Manager (p. 77)

Peristiwa Otomatisasi AWS Systems Manager

Notifikasi Perubahan Status Langkah Otomatisasi

```
{
  "version": "0",
  "id": "eeca120b-a321-433e-9635-dab369006a6b",
  "detail-type": "EC2 Automation Step Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-1:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-1:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "EndTime": "Nov 29, 2016 7:43:25 PM",
    "StartTime": "Nov 29, 2016 7:43:23 PM",
    "Time": 2630.0,
    "StepName": "runFixedCmds",
    "Action": "aws:runCommand"
  }
}
```

Notifikasi Perubahan Status Eksekusi Otomatisasi

```
{
  "version": "0",
  "id": "d290ece9-1088-4383-9df6-cd5b4ac42b99",
  "detail-type": "EC2 Automation Execution Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-1:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-1:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "StartTime": "Nov 29, 2016 7:43:20 PM",
    "EndTime": "Nov 29, 2016 7:43:26 PM",
    "Time": 5753.0,
    "ExecutedBy": "arn:aws:iam::123456789012:user/userName"
  }
}
```

Peristiwa Kalender Perubahan AWS Systems Manager

Berikut ini adalah contoh peristiwa untuk Kalender Perubahan AWS Systems Manager.

Note

Perubahan status untuk kalender yang dibagikan dari akun AWS lain tidak didukung.

Kalender BUKA

```
{
    "version": "0",
    "id": "47a3f03a-f30d-1011-ac9a-du3bdEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "111222333444",
    "time": "2020-09-19T18:00:07Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:111222333444:document/MyCalendar"
    ],
    "detail": {
        "state": "OPEN",
        "atTime": "2020-09-19T18:00:07Z",
        "nextTransitionTime": "2020-10-11T18:00:07Z"
    }
}
```

Kalender TUTUP

```
{
    "version": "0",
    "id": "f30df03a-1011-ac9a-47a3-f761eEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "111222333444",
    "time": "2020-09-17T21:40:02Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:111222333444:document/MyCalendar"
    ],
    "detail": {
        "state": "CLOSED",
        "atTime": "2020-08-17T21:40:00Z",
        "nextTransitionTime": "2020-09-19T18:00:07Z"
    }
}
```

Peristiwa Kepatuhan AWS Systems Manager

Berikut ini adalah contoh peristiwa untuk AWS Systems Manager.

Asosiasi Patuh

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:03:26Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
```

```
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Asosiasi Tidak Patuh

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:02:31Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "non_compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Patch Patuh

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:03:26Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

Patch Tidak Patuh

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
```

```
  "time": "2017-07-17T19:02:31Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "non_compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

Peristiwa Windows Maintenance AWS Systems Manager

Berikut ini adalah contoh dari peristiwa untuk Systems Manager Windows Maintenance.

Daftarkan Target

Nilai status valid lainnya adalah `DEREGISTERED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Target Registration Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T00:58:37Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:001312665065:maintenancewindow/mw-0ed7251d3fcf6e0c2",
      "arn:aws:ssm:us-west-2:001312665065:windowtarget/
e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6"
   ],
   "detail":{
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "status":"REGISTERED"
   }
}
```

Jenis Eksekusi Windows

Nilai status valid lainnya adalah `PENDING`, `IN_PROGRESS`, `SUCCESS`, `FAILED`, `TIMED_OUT`, dan `SKIPPED_OVERLAPPING`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Execution State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
```

```
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "status":"TIMED_OUT"
   }
}
```

Jenis Eksekusi Tugas

Nilai status valid lainnya adalah `IN_PROGRESS`, `SUCCESS`, `FAILED`, dan `TIMED_OUT`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Execution State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:56Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.759Z",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "end-time":"2016-11-16T01:00:56.847Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "status":"TIMED_OUT"
   }
}
```

Target Tugas Diproses

Nilai status valid lainnya adalah `IN_PROGRESS`, `SUCCESS`, `FAILED`, dan `TIMED_OUT`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Target Invocation State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-123456789012",
      "status":"TIMED_OUT",
      "owner-information":"Owner"
   }
}
```

Perubahan Status Windows

Nilai status valid adalah `ENABLED` dan `DISABLED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T00:58:37Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "window-id":"mw-123456789012",
      "status":"DISABLED"
   }
}
```

Peristiwa Menyimpan Parameter AWS Systems Manager

Berikut ini adalah contoh dari peristiwa untuk Systems Manager Menyimpan Parameter.

Buat Parameter

```
{
  "version": "0",
  "id": "6a7e4feb-b491-4cf7-a9f1-bf3703497718",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Create",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Perbarui Parameter

```
{
  "version": "0",
  "id": "9547ef2d-3b7e-4057-b6cb-5fdf09ee7c8f",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:44:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Update",
    "name": "foo",
```

```
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Hapus Parameter

```
{
  "version": "0",
  "id": "80e9b391-6a9b-413c-839a-453b528053af",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:45:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Delete",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Peristiwa Run Command AWS Systems Manager

Notifikasi Perubahan Status Run Command

```
{
    "version": "0",
    "id": "51c0891d-0e34-45b1-83d6-95db273d1602",
    "detail-type": "EC2 Command Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "expire-after": "2016-07-14T22:01:30.049Z",
        "parameters": {
            "executionTimeout": ["3600"],
            "commands": ["date"]
        },
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

Notifikasi Perubahan Status Pemanggilan Run Command

```
{
    "version": "0",
    "id": "4780e1b8-f56b-4de5-95f2-95db273d1602",
    "detail-type": "EC2 Command Invocation Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
```

```
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "instance-id": "i-9bb89e2b",
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

Peristiwa State Manager AWS Systems Manager

Perubahan Status Asosiasi State Manager

```
{
   "version":"0",
   "id":"db839caf-6f6c-40af-9a48-25b2ae2b7774",
   "detail-type":"EC2 State Manager Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-05-16T23:01:10Z",
   "region":"us-west-1",
   "resources":[
      "arn:aws:ssm:us-west-1::document/AWS-RunPowerShellScript"
   ],
   "detail":{
      "association-id":"6e37940a-23ba-4ab0-9b96-5d0a1a05464f",
      "document-name":"AWS-RunPowerShellScript",
      "association-version":"1",
      "document-version":"Optional.empty",
      "targets":"[{\"key\":\"InstanceIds\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2017-02-13T17:22:54.458Z",
      "last-successful-execution-date":"2017-05-16T23:00:01Z",
      "last-execution-date":"2017-05-16T23:00:01Z",
      "last-updated-date":"2017-02-13T17:22:54.458Z",
      "status":"Success",
      "association-status-aggregated-count":"{\"Success\":1}",
      "schedule-expression":"cron(0 */30 * * * ? *)",
      "association-cwe-version":"1.0"
   }
}
```

Perubahan Status Asosiasi Instans State Manager

```
{
   "version":"0",
   "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
   "detail-type":"EC2 State Manager Instance Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-02-23T15:23:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
      "arn:aws:ssm:us-east-1:123456789012:document/my-custom-document"
   ],
   "detail":{
      "association-id":"34fcb7e0-9a14-4984-9989-0e04e3f60bd8",
      "instance-id":"i-12345678",
      "document-name":"my-custom-document",
      "document-version":"1",
```

```
      "targets":"[{\"key\":\"instanceids\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2017-02-23T15:23:48Z",
      "last-successful-execution-date":"2017-02-23T16:23:48Z",
      "last-execution-date":"2017-02-23T16:23:48Z",
      "status":"Success",
      "detailed-status":"",
      "error-code":"testErrorCode",
      "execution-summary":"testExecutionSummary",
      "output-url":"sampleurl",
      "instance-association-cwe-version":"1"
   }
}
```

Peristiwa AWS Step Functions

Untuk contoh peristiwa Step Functions, lihat Contoh Peristiwa Step Functions dalam Panduan Developer AWS Step Functions.

Tandai Perubahan Peristiwa di Sumber Daya AWS

Berikut adalah contoh dari peristiwa penandaan.

```
{
    "version": "0",
    "id": "ffd8a6fe-32f8-ef66-c85c-111111111111",
    "detail-type": "Tag Change on Resource",
    "source": "aws.tag",
    "account": "123456789012",
    "time": "2018-09-18T20:41:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa"
    ],
    "detail": {
        "changed-tag-keys": [
            "key2",
            "key3"
        ],
        "service": "ec2",
        "resource-type": "instance",
        "version": 5,
        "tags": {
            "key4": "value4",
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

Peristiwa AWS Trusted Advisor

Berikut ini adalah contoh peristiwa untuk AWS Trusted Advisor. Untuk informasi selengkapnya, lihatPemantauanTrusted AdvisorPeriksa Hasil dengan AmazonCloudWatchPeristiwadiAWS SupportPanduan Pengguna.

Instans Amazon EC2 Penggunaan Rendah


```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T20:07:49Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "check-name": "Low Utilization Amazon EC2 Instances",
    "check-item-detail": {
      "Day 1": "0.1%  0.00MB",
      "Day 2": "0.1%  0.00MB",
      "Day 3": "0.1%  0.00MB",
      "Region/AZ": "ca-central-1a",
      "Estimated Monthly Savings": "$9.22",
      "14-Day Average CPU Utilization": "0.1%",
      "Day 14": "0.1%  0.00MB",
      "Day 13": "0.1%  0.00MB",
      "Day 12": "0.1%  0.00MB",
      "Day 11": "0.1%  0.00MB",
      "Day 10": "0.1%  0.00MB",
      "14-Day Average Network I/O": "0.00MB",
      "Number of Days Low Utilization": "14 days",
      "Instance Type": "t2.micro",
      "Instance ID": "i-01234567890abcdef",
      "Day 8": "0.1%  0.00MB",
      "Instance Name": null,
      "Day 9": "0.1%  0.00MB",
      "Day 4": "0.1%  0.00MB",
      "Day 5": "0.1%  0.00MB",
      "Day 6": "0.1%  0.00MB",
      "Day 7": "0.1%  0.00MB"
    },
    "status": "WARN",
    "resource_id": "arn:aws:ec2:ca-central-1:123456789012:instance/i-01234567890abcdef",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```


Optimalisasi Penyeimbang Beban


```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T20:07:03Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "check-name": "Load Balancer Optimization ",
    "check-item-detail": {
      "Instances in Zone a": "1",
      "Status": "Yellow",
      "Instances in Zone b": "0",
      "# of Zones": "2",
      "Region": "eu-central-1",
      "Load Balancer Name": "my-load-balance",
      "Instances in Zone e": null,
```

```
      "Instances in Zone c": null,
      "Reason": "Single AZ",
      "Instances in Zone d": null
    },
    "status": "WARN",
    "resource_id": "arn:aws:elasticloadbalancing:eu-central-1:123456789012:loadbalancer/my-
load-balancer",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```

Exposed Access Keys

```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T19:38:24Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "check-name": "Exposed Access Keys",
    "check-item-detail": {
      "Case ID": "12345678-1234-1234-abcd-1234567890ab",
      "Usage (USD per Day)": "0",
      "User Name (IAM or Root)": "my-username",
      "Deadline": "1440453299248",
      "Access Key ID": "AKIAIOSFODNN7EXAMPLE",
      "Time Updated": "1440021299248",
      "Fraud Type": "Exposed",
      "Location": "www.example.com"
    },
    "status": "ERROR",
    "resource_id": "",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```

WorkSpacesPeristiwa

Untuk informasi tentangWorkSpacesperistiwa, lihatMemantau AndaWorkSpacesMenggunakanCloudWatchPeristiwadiAmazonWorkSpacesPanduan Administrasi.

Peristiwa yang Disampaikan ViaCloudTrail

Anda juga dapat menggunakanCloudWatchPeristiwa dengan layanan yang tidak menghasilkan peristiwa dan tidak tercantum di halaman ini.AWS CloudTrailadalah layanan yang secara otomatis mencatat peristiwa sepertiAWSPanggilan API. Anda dapat membuatCloudWatchPeristiwa aturan yang memicu informasi yang ditangkap olehCloudTrail. Untuk informasi selengkapnya tentang CloudTrail, lihat Apa yang dimaksud dengan AWS CloudTrail?. Untuk informasi lebih lanjut tentang membuatCloudWatchAturan peristiwa yang menggunakanCloudTrail, lihatMembuat Aturan CloudWatch Events yang Memicu Panggilan API AWS Menggunakan AWS CloudTrail (p. 8).

Semua acara yang disampaikan melaluiCloudTrailmemilikiAWS API Call via CloudTrailsebagai nilai untukdetail-type.

Beberapa kejadian diAWSdapat dilaporkan keCloudWatchAcara baik oleh layanan itu sendiri dan olehCloudTrail, tetapi dengan cara yang berbeda. Misalnya, panggilan API Amazon EC2 yang meluncurkan atau mengakhiri instans menghasilkan peristiwa yang tersedia untukCloudWatchPeristiwa melaluiCloudTrail. Namun, perubahan keadaan instans Amazon EC2, dari 'berjalan' ke 'berakhir' misalnya, adalahCloudWatchAcara acara sendiri.

Berikut ini adalah contoh peristiwa yang disampaikan melaluiCloudTrail. Acara ini dihasilkan oleh Panggilan API AWS ke Amazon S3 untuk membuat bucket.

```
{
    "version": "0",
    "id": "36eb8523-97d0-4518-b33d-ee3579ff19f0",
    "detail-type": "AWS API Call via CloudTrail",
    "source": "aws.s3",
    "account": "123456789012",
    "time": "2016-02-20T01:09:13Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "Root",
            "principalId": "123456789012",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
                    "creationDate": "2016-02-20T01:05:59Z"
                }
            }
        },
        "eventTime": "2016-02-20T01:09:13Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "CreateBucket",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "100.100.100.100",
        "userAgent": "[S3Console/0.4]",
        "requestParameters": {
            "bucketName": "bucket-test-iad"
        },
        "responseElements": null,
        "requestID": "9D767BCC3B4E7487",
        "eventID": "24ba271e-d595-4e66-a7fd-9c16cbf8abae",
        "eventType": "AwsApiCall"
    }
}
```

Peristiwa panggilan API AWS dengan ukuran lebih besar dari 256 KB tidak didukung. Untuk informasi selengkapnya tentang panggilan API yang dapat Anda gunakan untuk aturan, lihatLayanan yang Didukung olehCloudTrailRiwayat Peristiwa.

Mengirim dan Menerima Peristiwa Antara Akun AWS

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat mengatur akun AWS untuk mengirim peristiwa ke akun AWS lain, atau untuk menerima peristiwa dari akun lain. Hal ini dapat berguna jika akun milik organisasi yang sama, atau milik organisasi yang merupakan mitra atau memiliki hubungan serupa.

Jika Anda mengatur akun untuk mengirim atau menerima peristiwa, Anda menentukan akun AWS mana yang dapat mengirim peristiwa ke atau menerima peristiwa dari Anda. Jika Anda menggunakan fitur AWS Organizations, Anda dapat menentukan organisasi dan memberikan akses ke semua akun di organisasi tersebut. Untuk informasi selengkapnya, lihat Apa itu AWS Organizations dalam Panduan Pengguna AWS Organizations.

Prosesnya adalah sebagai berikut:

- Pada akun penerima, edit izin pada bus peristiwa default untuk mengizinkan akun AWS yang ditentukan, organisasi, atau semua akun AWS untuk mengirim peristiwa ke akun penerima.
- Pada akun pengirim, atur satu atau lebih aturan dengan bus peristiwa default milik akun penerima sebagai target.

  Jika akun pengirim mewarisi izin untuk mengirim peristiwa karena merupakan bagian dari organisasi AWS yang memiliki izin, akun pengirim juga harus memiliki IAM role dengan kebijakan yang mengizinkannya untuk mengirim peristiwa ke akun penerima. Jika Anda menggunakan AWS Management Console untuk membuat aturan yang menargetkan akun penerima, peran dibuat secara otomatis. Jika Anda menggunakan AWS CLI, Anda harus membuat peran secara manual.
- Pada akun penerima, atur satu atau lebih aturan yang cocok dengan peristiwa yang berasal dari akun pengirim.

Parameter Wilayah AWS di mana akun penerima menambahkan izin untuk bus peristiwa default harus wilayah yang sama di mana akun pengirim menciptakan aturan untuk mengirim peristiwa ke akun penerima.

Peristiwa yang dikirim dari satu akun ke akun lainnya dibebankan ke akun pengiriman sebagai peristiwa kustom. Akun penerima tidak dikenakan biaya. Untuk informasi selengkapnya, lihat Harga Amazon CloudWatch.

Jika akun penerima menetapkan aturan yang mengirimkan peristiwa yang diterima dari akun pengirim ke akun ketiga, peristiwa tersebut tidak dikirim ke akun ketiga.

Mengaktifkan Akun AWS untuk Menerima Peristiwa dari Akun AWS Lainnya

Untuk menerima peristiwa dari akun atau organisasi lain, Anda harus terlebih dahulu mengedit izin pada bus peristiwa default akun Anda. Bus peristiwa default menerima peristiwa dari layanan AWS, akun AWSterotorisasi lainnya, dan panggilan `PutEvents`.

Ketika Anda mengedit izin pada bus peristiwa default Anda untuk memberikan izin untuk akun AWS lainnya, Anda dapat menentukan akun berdasarkan ID akun atau ID organisasi. Atau Anda dapat memilih untuk menerima peristiwa dari semua akun AWS.

> Warning
>
> Jika Anda memilih untuk menerima peristiwa dari semua akun AWS, hati-hati untuk membuat aturan yang hanya cocok dengan peristiwa untuk menerima dari lainnya. Untuk membuat aturan yang lebih aman, pastikan bahwa pola peristiwa untuk setiap aturan berisi bidang `Account` dengan ID akun dari satu atau lebih akun yang darinya untuk menerima peristiwa. Aturan yang memiliki pola peristiwa yang berisi bidang Akun tidak cocok dengan peristiwa yang dikirim dari akun yang tidak tercantum dalam bidang `Account`. Untuk informasi selengkapnya, lihat Pola Peristiwa di CloudWatch Events (p. 37).

Untuk mengaktifkan akun Anda untuk menerima peristiwa dari akun AWS lainnya yang menggunakan konsol

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Bus Peristiwa, Tambahkan Izin.
3. Pilih Akun AWS atau Organisasi.

   Jika Anda memilih Akun AWS, masukkan 12 digit ID akun AWS dari akun untuk menerima peristiwa. Untuk menerima peristiwa dari semua akun AWS lain, pilih Semua orang(*).

   Jika Anda memilih Organisasi, pilih Organisasi saya untuk memberikan izin ke semua akun dalam organisasi yang salah satu anggotanya adalah akun saat ini. Atau pilih Organisasi lain dan masukkan ID organisasi dari organisasi tersebut. Anda harus menyertakan awalan `o-` saat Anda mengetik ID organisasi.
4. Pilih Tambahkan.
5. Anda dapat mengulangi langkah-langkah ini untuk menambahkan akun atau organisasi lain.

Untuk mengaktifkan akun Anda untuk menerima peristiwa dari akun AWS lainnya yang menggunakan AWS CLI

1. Untuk mengaktifkan satu akun AWS spesifik untuk mengirim peristiwa, jalankan perintah berikut:

```
aws events put-permission --action events:PutEvents --statement-id MySid --
principal SenderAccountID
```

   Untuk mengaktifkan organisasi AWS untuk mengirim peristiwa, jalankan perintah berikut:

```
aws events put-permission --action events:PutEvents --statement-id MySid
                  --principal \* --condition '{"Type" : "StringEquals", "Key":
 "aws:PrincipalOrgID", "Value": "SenderOrganizationID"}'
```

   Untuk mengaktifkan semua akun AWS lainnya untuk mengirim peristiwa, jalankan perintah berikut:

```
aws events put-permission --action events:PutEvents --statement-id MySid --principal \*
```

Anda dapat menjalankan `aws events put-permission` beberapa kali untuk memberikan izin untuk kedua akun AWS individu dan organisasi, tetapi Anda tidak dapat menentukan akun individu dan organisasi dalam satu perintah.

2. Setelah menetapkan izin untuk bus peristiwa default Anda, Anda dapat secara opsional menggunakan perintah `describe-event-bus` untuk memeriksa izin:

```
aws events describe-event-bus
```

Mengirim Peristiwa ke Akun AWS Lain

Untuk mengirim peristiwa ke akun lain, konfigurasikan aturan CloudWatch Events yang memiliki bus peristiwa default akun AWS lain sebagai target. Bus peristiwa default yang menerima akun juga harus dikonfigurasi untuk menerima peristiwa dari akun Anda.

Untuk mengirim peristiwa dari akun Anda ke akun AWS lainnya yang menggunakan konsol

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat Aturan.
3. Untuk Sumber Peristiwa, pilih Pola Peristiwa dan pilih nama layanan dan jenis peristiwa untuk dikirim ke akun lain.
4. Pilih Tambahkan Widget.
5. Untuk Target, pilih Bus peristiwa di akun AWS lain. Untuk ID Akun, masukkan 12 digit ID akun AWS dari akun untuk mengirim peristiwa.
6. IAM role diperlukan ketika akun pengirim ini memiliki izin untuk mengirim peristiwa karena akun penerima memberikan izin kepada seluruh organisasi.

   - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Jika tidak, pilih Gunakan peran yang ada. Pilih peran yang sudah memiliki izin yang memadai untuk meminta pembangunan ini. CloudWatch Events tidak memberikan izin tambahan untuk peran yang Anda pilih.
7. Di bagian bawah halaman, pilih Konfigurasi Detail.
8. Ketikkan sebuah nama dan deskripsi untuk aturan, lalu pilih Buat Aturan.

Untuk mengirim peristiwa ke akun AWS lain menggunakan AWS CLI

1. Jika akun pengirim mewarisi izin untuk mengirim peristiwa karena merupakan bagian dari organisasi AWS yang memiliki izin dari akun penerima, akun pengirim juga harus memiliki peran dengan kebijakan yang mengizinkannya untuk mengirim peristiwa ke akun penerima. Langkah ini menjelaskan cara membuat peran tersebut.

   Jika akun pengirim diberikan izin untuk mengirim peristiwa melalui ID akun AWS, dan tidak melalui organisasi, langkah ini opsional. Anda bisa melewati langkah 2.

   a. Jika akun pengirim diberikan izin melalui organisasi, buat IAM role yang diperlukan. Pertama, buat file bernama `assume-role-policy-document.json` dengan isi berikut:

```
{
```

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

b. Jalankan perintah berikut untuk membuat peran:

```
$ aws iam create-role \
--profile sender \
--role-name event-delivery-role \
--assume-role-policy-document file://assume-role-policy-document.json
```

c. Buat file bernama `permission-policy.json` dengan konten berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:${receiver_account_id}:event-bus/default"
      ]
    }
  ]
}
```

d. Masukkan perintah berikut untuk melampirkan kebijakan ke peran:

```
$ aws iam put-role-policy \
--profile sender \
--role-name event-delivery-role \
--policy-name EventBusDeliveryRolePolicy
--policy-document file://permission-policy.json
```

2. Gunakan perintah `put-rule` untuk membuat aturan yang cocok dengan jenis peristiwa untuk dikirim ke akun lain.
3. Tambahkan bus peristiwa default akun lain sebagai target aturan.

   Jika akun pengirim diberi izin untuk mengirim peristiwa dengan ID akunnya, peran tidak perlu ditentukan. Jalankan perintah berikut:

```
aws events put-targets --rule NameOfRuleMatchingEventsToSend --targets
 "Id"="MyId","Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/default"
```

   Jika akun pengirim diberi izin untuk mengirim peristiwa oleh organisasinya, tentukan peran, seperti dalam contoh berikut:

```
aws events put-targets --rule NameOfRuleMatchingEventsToSend --targets
 "Id"="MyId","Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/
default","RoleArn"="arn:aws:iam:${sender_account_id}:role/event-delivery-role"
```

Menulis Aturan yang Cocok dengan Peristiwa dari Akun AWS Lain

Jika akun Anda diatur untuk menerima peristiwa dari akun AWS lainnya, Anda dapat menulis aturan yang cocok dengan peristiwa tersebut. Atur pola peristiwa aturan agar cocok dengan peristiwa yang Anda terima dari akun lain.

Kecuali Anda menentukan `account` dalam pola peristiwa aturan, salah satu aturan akun Anda, baik yang baru maupun yang sudah ada, yang cocok dengan peristiwa yang Anda terima dari pemicu akun lain berdasarkan peristiwa tersebut. Jika Anda menerima peristiwa dari akun lainnya, dan Anda ingin aturan untuk memicu hanya pada pola peristiwa ketika dibuat dari akun Anda sendiri, Anda harus menambahkan `account` dan menentukan ID akun Anda sendiri ke pola peristiwa aturan.

Jika Anda mengatur akun AWS untuk menerima peristiwa dari semua akun AWS, kami sangat menyarankan agar Anda menambahkan `account` ke setiap aturan CloudWatch Events di akun Anda. Hal ini mencegah aturan di akun Anda memicu peristiwa dari akun AWS yang tidak dikenal. Saat Anda menentukan bidang `account` dalam aturan, Anda dapat menentukan ID akun dari lebih dari satu akun AWS di bidang.

Untuk memiliki pemicu aturan pada peristiwa yang cocok dari akun AWS apa pun yang telah Anda berikan izin, jangan menentukan* di bidang `account` aturan. Melakukannya tidak akan cocok dengan peristiwa apa pun, karena * tidak pernah muncul di bidang `account` peristiwa. Alih-alih, hanya menghilangkan bidang `account` dari aturan.

Untuk menulis aturan pencocokan peristiwa dari akun lain menggunakan konsol

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Peristiwa, Buat Aturan.
3. Untuk Sumber Peristiwa, pilih Pola Peristiwa dan pilih nama layanan dan jenis peristiwa yang sesuai aturan.
4. Dekat Pola Pratinjau Peristiwa, pilih Edit.
5. Di jendela edit, tambahkan baris `Account` yang menentukan mana akun AWS pengirim peristiwa ini yang harus dicocokkan dengan aturan. Sebagai contoh, jendela edit awalnya menunjukkan hal berikut:

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ]
}
```

Tambahkan berikut ini untuk agar aturan cocok dengan pemberitahuan volume EBS yang dikirim oleh akun AWS 123456789012 dan 111122223333:

```
{
  "account": [
    "123456789012","111122223333"
  ],
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ]
```

```
}
```

6. Setelah mengedit pola peristiwa, pilih Simpan.
7. Selesaikan pembuatan aturan seperti biasa, tetapkan satu atau lebih target di akun Anda.

Untuk menulis aturan pencocokan peristiwa dari akun AWS lain menggunakan AWS CLI

- Gunakan perintah `put-rule`. Di field `Account` dalam pola peristiwa aturan, tentukan akun AWS lain agar sesuai aturan. Contoh aturan berikut cocok dengan perubahan status instans Amazon EC2 di akun AWS 123456789012 dan 111122223333:

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"account\":
["123456789012", "111122223333"],\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2
 Instance State-change Notification\"]}"  --role-arn "arn:aws:iam::123456789012:role/
MyRoleForThisRule"
```

Memigrasi Hubungan Pengirim-Penerima untuk Menggunakan AWS Organizations

Jika Anda memiliki akun pengirim yang izinnya diberikan langsung ke akun ID, dan Anda sekarang ingin mencabut izin tersebut dan memberikan akses akun pengiriman dengan memberikan izin kepada organisasi, Anda harus mengambil beberapa langkah tambahan. Langkah-langkah ini memastikan bahwa peristiwa dari akun pengirim masih bisa menuju akun penerima. Hal ini karena akun yang diberikan izin untuk mengirim peristiwa melalui organisasi juga harus menggunakan IAM role untuk melakukannya.

Untuk menambahkan izin yang diperlukan untuk memigrasi hubungan penerima pengirim

1. Di akun pengirim, buat IAM role dengan kebijakan yang memungkinkannya untuk mengirim peristiwa ke akun penerima.

   a. Buat file bernama `assume-role-policy-document.json` dengan konten berikut:

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

   b. Jalankan perintah berikut untuk membuat IAM role:

```
$ aws iam create-role \
--profile sender \
--role-name event-delivery-role \
--assume-role-policy-document file://assume-role-policy-document.json
```

   c. Buat file bernama `permission-policy.json` dengan konten berikut:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:${receiver_account_id}:event-bus/default"
      ]
    }
  ]
}
```

d. Masukkan perintah berikut untuk melampirkan kebijakan ke peran:

```
$ aws iam put-role-policy \
--profile sender \
--role-name event-delivery-role \
--policy-name EventBusDeliveryRolePolicy
--policy-document file://permission-policy.json
```

2. Mengedit setiap aturan yang ada di akun pengirim dengan bus peristiwa default akun penerima sebagai target. Mengedit aturan dengan menambahkan peran yang Anda buat di langkah 1 ke informasi target. Gunakan perintah berikut:

```
aws events put-targets --rule Rulename --targets
 "Id"="MyID","Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/
default","RoleArn"="arn:aws:iam:${sender_account_id}:role/event-delivery-role"
```

3. Di account penerima, jalankan perintah berikut untuk memberikan izin untuk akun di organisasi untuk mengirim peristiwa ke akun penerima:

```
aws events put-permission --action events:PutEvents --statement-id Sid-For-Organization
 --principal \* --condition '{"Type" : "StringEquals", "Key": "aws:PrincipalOrgID",
 "Value": "SenderOrganizationID"}'
```

Secara opsional, Anda juga dapat mencabut izin yang awalnya diberikan langsung ke akun pengirim:

```
aws events remove-permission --statement-id Sid-for-SenderAccount
```

Menambahkan Peristiwa dengan PutEvents

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Tindakan `PutEvents` mengirimkan beberapa peristiwa ke CloudWatch Events dalam permintaan tunggal. Untuk informasi selengkapnya, lihat PutEvents dalam Referensi API Amazon CloudWatch Events dan put-events dalam Referensi Perintah AWS CLI.

Setiap permintaan `PutEvents` dapat mendukung sejumlah entri yang terbatas. Untuk informasi selengkapnya, lihat Kuota CloudWatch Events (p. 114). Operasi `PutEvents` mencoba untuk memproses semua entri dalam urutan alami permintaan. Setiap peristiwa memiliki id unik yang ditetapkan oleh CloudWatch Events setelah Anda memanggil `PutEvents`.

Contoh kode Java berikut ini mengirimkan dua peristiwa yang identik ke CloudWatch Events:

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

PutEventsRequest request = new PutEventsRequest()
        .withEntries(requestEntry, requestEntry);

PutEventsResult result = awsEventsClient.putEvents(request);

for (PutEventsResultEntry resultEntry : result.getEntries()) {
    if (resultEntry.getEventId() != null) {
        System.out.println("Event Id: " + resultEntry.getEventId());
    } else {
        System.out.println("Injection failed with Error Code: " +
 resultEntry.getErrorCode());
    }
}
```

Hasil `PutEvents` mencakup susunan entri respons. Setiap entri dalam susunan respons secara langsung berkorelasi dengan entri dalam susunan permintaan dengan urutan asal, dari permintaan dan respons paling atas hingga paling bawah. Array `Entries` respons selalu mencakup jumlah entri yang sama sebagai array permintaan.

Menangani Kegagalan Saat Menggunakan PutEvents

Secara default, kegagalan tiap-tiap entri dalam permintaan tidak menghentikan pemrosesan entri berikutnya dalam permintaan. Ini berarti bahwa susunan Entri respons mencakup entri yang berhasil maupun tidak berhasil diproses. Anda harus mendeteksi entri yang tidak berhasil diproses dan menyertakannya ke dalam panggilan berikutnya.

Entri hasil yang berhasil mencakup nilai Id, dan entri hasil yang tidak berhasil mencakup nilai `ErrorCode` dan `ErrorMessage`. Parameter `ErrorCode` mencerminkan jenis kesalahan. `ErrorMessage` memberikan informasi lebih terperinci tentang kesalahan tersebut. Contoh di bawah ini memiliki tiga entri hasil untuk permintaan `PutEvents`. Entri kedua telah gagal dan tercermin dalam respons.

Contoh: Sintaks Respons PutEvents

```
{
    "FailedEntryCount": 1,
    "Entries": [
        {
            "EventId": "11710aed-b79e-4468-a20b-bb3c0c3b4860"
        },
        {   "ErrorCode": "InternalFailure",
            "ErrorMessage": "Internal Service Failure"
        },
            "EventId": "d804d26a-88db-4b66-9eaf-9a11c708ae82"
        }
    ]
}
```

Entri yang tidak berhasil diproses dapat disertakan dalam permintaan `PutEvents` berikutnya. Pertama, periksa parameter `FailedRecordCount` dalam `PutEventsResult` untuk mengonfirmasi apakah ada catatan kegagalan dalam permintaan. Jika ada, tiap-tiap `Entry` dengan `ErrorCode` bukan null harus ditambahkan ke permintaan berikutnya. Untuk contoh jenis handler ini, lihat kode berikut.

Contoh: Handler kegagalan PutEvents

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

List<PutEventsRequestEntry> putEventsRequestEntryList = new ArrayList<>();
for (int i = 0; i < 3; i++) {
    putEventsRequestEntryList.add(requestEntry);
}

PutEventsRequest putEventsRequest = new PutEventsRequest();
putEventsRequest.withEntries(putEventsRequestEntryList);
PutEventsResult putEventsResult = awsEventsClient.putEvents(putEventsRequest);

while (putEventsResult.getFailedEntryCount() > 0) {
    final List<PutEventsRequestEntry> failedEntriesList = new ArrayList<>();
    final List<PutEventsResultEntry> PutEventsResultEntryList =
 putEventsResult.getEntries();
    for (int i = 0; i < PutEventsResultEntryList.size(); i++) {
        final PutEventsRequestEntry putEventsRequestEntry =
 putEventsRequestEntryList.get(i);
```

```
        final PutEventsResultEntry putEventsResultEntry = PutEventsResultEntryList.get(i);
        if (putEventsResultEntry.getErrorCode() != null) {
            failedEntriesList.add(putEventsRequestEntry);
        }
    }
    putEventsRequestEntryList = failedEntriesList;
    putEventsRequest.setEntries(putEventsRequestEntryList);
    putEventsResult = awsEventsClient.putEvents(putEventsRequest);
    }
```

Mengirim Peristiwa Menggunakan AWS CLI

Anda dapat menggunakan AWS CLI untuk mengirim peristiwa kustom. Contoh berikut ini menempatkan satu peristiwa kustom ke CloudWatch Events:

```
aws events put-events \
--entries '[{"Time": "2016-01-14T01:02:03Z", "Source": "com.mycompany.myapp", "Resources":
 ["resource1", "resource2"], "DetailType": "myDetailType", "Detail": "{ \"key1\":
 \"value1\", \"key2\": \"value2\" }"}]'
```

Anda juga dapat membuat file misalnya, entries.json, seperti berikut ini:

```
[
  {
    "Time": "2016-01-14T01:02:03Z",
    "Source": "com.mycompany.myapp",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }"
  }
]
```

Anda dapat menggunakan AWS CLI untuk membaca entri dari file ini dan mengirim peristiwa. Di jendela perintah, ketik:

```
aws events put-events --entries file://entries.json
```

Menghitung Ukuran Entri Peristiwa PutEvents

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat memasukkan peristiwa kustom ke CloudWatch Events menggunakan tindakan `PutEvents`. Anda dapat memasukkan beberapa peristiwa menggunakan tindakan `PutEvents` selama ukuran entri total kurang dari 256 KB. Anda dapat menghitung ukuran entri peristiwa terlebih dahulu dengan mengikuti

langkah-langkah di bawah ini. Anda dapat membuat batch berisi beberapa entri peristiwa menjadi satu permintaan agar efisien.

> **Note**
>
> Batas ukuran dikenakan pada entri. Meskipun entri kurang dari batas ukuran tersebut, bukan berarti peristiwa di CloudWatch Events juga kurang dari ukuran ini. Sebaliknya, ukuran peristiwa selalu lebih besar dari ukuran entri akibat karakter yang diperlukan dan kunci representasi JSON peristiwa. Untuk informasi selengkapnya, lihat Pola Peristiwa di CloudWatch Events (p. 37).

Ukuran `PutEventsRequestEntry` dihitung sebagai berikut:

- Jika parameter `Time` ditentukan, maka ukurannya sebesar 14 bit.
- Parameter `Source` dan `DetailType` adalah ukuran jumlah bit untuk bentuknya yang berenkode UTF-8.
- Jika parameter `Detail` ditentukan, ukurannya adalah jumlah bit untuk bentuknya yang berenkode UTF-8.
- Jika parameter `Resources` ditentukan, ukuran setiap entri adalah jumlah bit untuk bentuknya yang berenkode UTF-8.

Contoh kode Java berikut ini menghitung ukuran objek `PutEventsRequestEntry` yang diberikan:

```
int getSize(PutEventsRequestEntry entry) {
    int size = 0;
    if (entry.getTime() != null) {
        size += 14;
    }
    size += entry.getSource().getBytes(StandardCharsets.UTF_8).length;
    size += entry.getDetailType().getBytes(StandardCharsets.UTF_8).length;
    if (entry.getDetail() != null) {
        size += entry.getDetail().getBytes(StandardCharsets.UTF_8).length;
    }
    if (entry.getResources() != null) {
        for (String resource : entry.getResources()) {
            if (resource != null) {
                size += resource.getBytes(StandardCharsets.UTF_8).length;
            }
        }
    }
    return size;
}
```

Menggunakan CloudWatch Events dengan VPC Endpoint Antarmuka

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meng-hosting sumber daya AWS, Anda dapat membuat koneksi privat antara VPC dan CloudWatch Events Anda. Anda dapat menggunakan koneksi ini untuk mengaktifkan CloudWatch Events untuk berkomunikasi dengan sumber daya di VPC Anda tanpa melalui internet publik.

Amazon VPC adalah layanan AWSyang dapat Anda gunakan untuk meluncurkan sumber daya AWS dalam jaringan virtual yang Anda tetapkan. Dengan VPC, Anda memiliki kendali terhadap pengaturan jaringan, seperti rentang alamat IP, subnet, tabel rute, dan pintu masuk jaringan. Untuk menghubungkan VPC ke CloudWatch Events, tetapkan VPC endpoint antarmuka untuk CloudWatch Events. Jenis titik akhir ini memungkinkan Anda untuk menghubungkan VPC Anda ke layanan AWS. Titik akhir ini memberikan konektivitas andal dan dapat diatur skalanya ke CloudWatch Events tanpa memerlukan gateway internet, instans terjemahan alamat jaringan (NAT), atau koneksi VPN. Untuk informasi selengkapnya, lihat Apa yang dimaksud dengan Amazon VPC dalam Panduan Pengguna Amazon VPC.

VPC endpoint antarmuka didukung oleh AWS, PrivateLink, teknologi AWS yang mendukung komunikasi privat antara layanan AWS menggunakan antarmuka jaringan elastis dengan alamat IP privat. Untuk informasi selengkapnya, lihat Baru – AWS PrivateLink untuk Layanan AWS.

Langkah-langkah berikut ditujukan untuk pengguna Amazon VPC. Untuk informasi selengkapnya, lihat Memulai dalam Panduan Pengguna Amazon VPC.

Ketersediaan

CloudWatch Events saat ini mendukung VPC endpoint di Wilayah berikut:

- US East (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)

- Europe (Ireland)
- Europe (London)
- Eropa (Paris)
- South America (São Paulo)

Membuat VPC Endpoint untuk CloudWatch Events

Untuk mulai menggunakan CloudWatch Events dengan VPC Anda, buat sebuah VPC endpoint antarmuka untuk CloudWatch Events. Nama layanan yang harus dipilih adalah com.amazonaws.***Region***.events. Untuk informasi selengkapnya, lihat Membuat sebuah Titik Akhir Antarmuka dalam Panduan Pengguna Amazon VPC.

Anda tidak perlu mengubah pengaturan untuk CloudWatch Events. CloudWatch Events memanggil layanan AWS lain menggunakan titik akhir publik atau VPC endpoint antarmuka privat, mana pun yang sedang digunakan. Misalnya, jika Anda membuat VPC endpoint antarmuka untuk CloudWatch Events, dan Anda sudah memiliki aturan CloudWatch Events yang mengirimkan notifikasi ke Amazon SNS saat dipicu, notifikasi mulai mengalir melalui VPC endpoint antarmuka.

Mengendalikan Akses ke VPC Endpoint CloudWatch Events

Kebijakan titik akhir VPC adalah kebijakan sumber daya IAM yang Anda lampirkan ke titik akhir ketika membuat atau memodifikasi titik akhir. Jika Anda tidak melampirkan kebijakan ketika membuat titik akhir, kami melampirkan kebijakan default untuk Anda sehingga memungkinkan akses penuh ke layanan. Kebijakan titik akhir tidak membatalkan atau mengganti kebijakan pengguna IAM atau kebijakan khusus layanan. Ini adalah kebijakan terpisah untuk mengendalikan akses dari titik akhir ke layanan tertentu.

Kebijakan titik akhir harus ditulis dalam format JSON.

Untuk informasi selengkapnya, lihat Mengendalikan Akses ke Layanan dengan VPC Endpoint dalam Panduan Pengguna Amazon VPC.

Berikut adalah contoh kebijakan titik akhir untuk CloudWatch Events. Kebijakan ini memungkinkan pengguna terhubung ke CloudWatch Events melalui VPC untuk mengirimkan peristiwa ke CloudWatch Events dan mencegahnya melakukan tindakan CloudWatch Events lainnya.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "events:PutEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Untuk mengubah kebijakan VPC endpoint untuk CloudWatch Events

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.

2. Di panel navigasi, pilih Titik akhir.
3. Jika Anda belum membuat titik akhir untuk CloudWatch Events, pilih Buat Titik Akhir. Kemudian pilih com.amazonaws.***Region***.events dan pilih Buat titik akhir.
4. Pilih titik akhir com.amazonaws.***Region***.events, dan pilih tab Kebijakan di bagian bawah layar.
5. Pilih Edit Kebijakan dan buat perubahan pada kebijakan.

Pemantauan Penggunaan dengan Metrik CloudWatch

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

CloudWatch Events mengirimkan metrik ke Amazon CloudWatch setiap menitnya.

Metrik CloudWatch Events

`AWS/Events` Namespace mencakup metrik berikut.

Semua metrik ini menggunakan Count sebagai unit, sehingga Sum dan SampleCount adalah statistik yang paling berguna.

Metrik	Deskripsi
`DeadLetterInvocations`	Mengukur frekuensi target aturan tidak dipanggil sebagai tanggapan suatu peristiwa. Termasuk permohonan yang memicu kembali aturan yang sama, menyebabkan putaran tidak terbatas. Dimensi valid: ruleName Unit: Count
`Invocations`	Mengukur frekuensi target dipanggil untuk aturan sebagai tanggapan suatu peristiwa. Termasuk permohonan yang berhasil dan gagal, tetapi tidak termasuk upaya yang terhambat atau dicoba lagi hingga gagal secara permanen. Ini tidak termasuk DeadLetterInvocations. **Note** CloudWatch Events hanya mengirimkan metrik ini ke CloudWatch jika nilainya bukan nol. Dimensi valid: ruleName Unit: Count
`FailedInvocations`	Mengukur jumlah permohonan yang gagal secara permanen. Tidak termasuk permohonan yang diulang atau berhasil setelah dicoba lagi. Metrik ini juga tidak memperhitungkan permohonan gagal dalam DeadLetterInvocations. Dimensi valid: ruleName Unit: Count

Metrik	Deskripsi
`TriggeredRules`	Mengukur jumlah aturan yang dipicu yang sesuai dengan semua peristiwa. Dimensi valid: ruleName Unit: Count
`MatchedEvents`	Mengukur jumlah peristiwa yang sesuai dengan semua aturan. Dimensi valid: Tidak ada Unit: Count
`ThrottledRules`	Mengukur jumlah aturan yang dipicu yang sedang terhambat. Dimensi valid: ruleName Unit: Count

Dimensi untuk Metrik CloudWatch Events

Metrik CloudWatch Events memiliki satu dimensi, yang tercantum di bawah ini.

Dimensi	Deskripsi
`RuleName`	Menyaring metrik yang tersedia berdasarkan nama aturan.

Aturan Terkelola Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Layanan AWS dapat membuat dan mengelola aturan CloudWatch Events di akun AWS Anda yang diperlukan untuk fungsi-fungsi tertentu dalam layanan tersebut. Ini semuanya disebut aturan terkelola.

Ketika layanan membuat aturan terkelola, layanan juga dapat membuat kebijakan IAM yang memberikan izin ke layanan itu untuk membuat aturan. Kebijakan IAM yang dibuat dengan cara ini memiliki jangkauan sempit dengan perizinan tingkat sumber daya untuk mengizinkan penciptaan aturan yang diperlukan saja.

Anda dapat menghapus aturan terkelola menggunakan pilihan Hapus paksa. Lakukan hanya jika Anda yakin bahwa layanan lain tidak lagi membutuhkan aturan tersebut. Jika tidak, menghapus aturan terkelola menyebabkan fitur yang bergantung padanya berhenti bekerja.

Menggunakan CloudWatch Peristiwa denganAWSSDK

AWSkit pengembangan perangkat lunak (SDK) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan pengembang untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK for C++	AWS SDK for C++Contoh kode
AWS SDK for Go	AWS SDK for GoContoh kode
AWS SDK for Java	AWS SDK for JavaContoh kode
AWS SDK for JavaScript	AWS SDK for JavaScriptContoh kode
AWS SDK for .NET	AWS SDK for .NETContoh kode
AWS SDK for PHP	AWS SDK for PHPContoh kode
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3)Contoh kode
AWS SDK for Ruby	AWS SDK for RubyContoh kode

Untuk contoh khusus untuk CloudWatch Acara, lihatContoh kode untuk CloudWatch Peristiwa menggunakanAWSSDK (p. 100).

Contoh ketersediaan

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode dengan menggunakanBerikan umpan baliktautan di bagian bawah halaman ini.

Contoh kode untuk CloudWatch Peristiwa menggunakanAWSSDK

Contoh kode berikut ini menunjukkan cara menggunakan CloudWatch Peristiwa denganAWSkit pengembangan perangkat lunak (SDK).

Contoh dibagi ke dalam kategori berikut:

Tindakan

Kutipan kode yang menunjukkan cara memanggil fungsi layanan individual.

Untuk daftar lengkapAWSPanduan pengembang SDK dan contoh kode, lihatMenggunakan CloudWatch Peristiwa denganAWSSDK (p. 99). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Tindakan untuk CloudWatch Peristiwa menggunakanAWSSDK (p. 100)
    - Tambahkan sebuah target fungsi Lambda menggunakanAWSSDK (p. 100)
    - Buat CloudWatch Aturan yang dijadwalkan peristiwa menggunakanAWSSDK (p. 103)
    - mengirim CloudWatch Acara peristiwa menggunakanAWSSDK (p. 105)

Tindakan untuk CloudWatch Peristiwa menggunakanAWSSDK

Contoh kode berikut ini mendemonstrasikan cara melakukan individu CloudWatch Tindakan Peristiwa denganAWSSDK SDK. Kutipan ini memanggil CloudWatch Events API dan tidak dimaksudkan untuk dijalankan secara terpisah. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkap, lihatAmazon CloudWatch Referensi API Peristiwa.

Contoh

- Tambahkan sebuah target fungsi Lambda menggunakanAWSSDK (p. 100)
- Buat CloudWatch Aturan yang dijadwalkan peristiwa menggunakanAWSSDK (p. 103)
- mengirim CloudWatch Acara peristiwa menggunakanAWSSDK (p. 105)

Tambahkan sebuah target fungsi Lambda menggunakanAWSSDK

Contoh kode berikut ini menunjukkan cara menambahkanAWS Lambdatarget fungsi ke Amazon CloudWatch Peristiwa event.

Java

SDK for Java 2.x

Tip

Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

```
   public static void putCWTargets(CloudWatchEventsClient cwe, String ruleName,
String functionArn, String targetId ) {

       try {
           Target target = Target.builder()
               .arn(functionArn)
               .id(targetId)
               .build();

           PutTargetsRequest request = PutTargetsRequest.builder()
               .targets(target)
               .rule(ruleName)
               .build();

           cwe.putTargets(request);
           System.out.printf(
               "Successfully created CloudWatch events target for rule %s",
               ruleName);

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Untuk rincian selengkapnya API, lihatPutTargetsdiAWS SDK for Java 2.xReferensi API.

JavaScript

SDK for SDK for JavaScript V3

Tip

Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

Buat klien dalam modul terpisah dan ekspor.

```
import { CloudWatchEventsClient } from "@aws-sdk/client-cloudwatch-events";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon CloudWatch service client object.
export const cweClient = new CloudWatchEventsClient({ region: REGION });
```

Impor modul SDK dan klien dan panggil API.

```
// Import required AWS SDK clients and commands for Node.js
import { PutTargetsCommand } from "@aws-sdk/client-cloudwatch-events";
import { cweClient } from "./libs/cloudWatchEventsClient.js";
```

```
// Set the parameters
export const params = {
  Rule: "DEMO_EVENT",
  Targets: [
    {
      Arn: "LAMBDA_FUNCTION_ARN", //LAMBDA_FUNCTION_ARN
      Id: "myCloudWatchEventsTarget",
    },
  ],
};

export const run = async () => {
  try {
    const data = await cweClient.send(new PutTargetsCommand(params));
    console.log("Success, target added; requestID: ", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
// Uncomment this line to run execution within this file.
// run();
```

- Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK for JavaScript.
- Untuk rincian selengkapnya API, lihatPutTargetsdiAWS SDK for JavaScriptReferensi API.

SDK for SDK for JavaScript V2

> **Tip**
>
> Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'REGION'});

// Create CloudWatchEvents service object
var cwevents = new AWS.CloudWatchEvents({apiVersion: '2015-10-07'});

var params = {
  Rule: 'DEMO_EVENT',
  Targets: [
    {
      Arn: 'LAMBDA_FUNCTION_ARN',
      Id: 'myCloudWatchEventsTarget',
    }
  ]
};

cwevents.putTargets(params, function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK for JavaScript.
- Untuk rincian selengkapnya API, lihatPutTargetsdiAWS SDK for JavaScriptReferensi API.

Untuk daftar lengkapAWSPanduan pengembang SDK dan contoh kode, lihatMenggunakan CloudWatch Peristiwa denganAWSSDK (p. 99). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Buat CloudWatch Aturan yang dijadwalkan peristiwa menggunakanAWSSDK

Contoh kode berikut ini menunjukkan cara membuat Amazon CloudWatch Aturan acara dijadwalkan.

Java

SDK for Java 2.x

Tip

Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

```
   public static void putCWRule(CloudWatchEventsClient cwe, String ruleName,
String roleArn) {

       try {
           PutRuleRequest request = PutRuleRequest.builder()
               .name(ruleName)
               .roleArn(roleArn)
               .scheduleExpression("rate(5 minutes)")
               .state(RuleState.ENABLED)
               .build();

           PutRuleResponse response = cwe.putRule(request);
           System.out.printf(
                   "Successfully created CloudWatch events rule %s with arn %s",
                   roleArn, response.ruleArn());

       } catch (
           CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Untuk rincian selengkapnya API, lihatPutRulediAWS SDK for Java 2.xReferensi API.

JavaScript

SDK for SDK for JavaScript V3

Tip

Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

Buat klien dalam modul terpisah dan ekspor.

```
import { CloudWatchEventsClient } from "@aws-sdk/client-cloudwatch-events";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon CloudWatch service client object.
export const cweClient = new CloudWatchEventsClient({ region: REGION });
```

Impor modul SDK dan klien dan panggil API.

```
// Import required AWS SDK clients and commands for Node.js
import { PutRuleCommand } from "@aws-sdk/client-cloudwatch-events";
import { cweClient } from "./libs/cloudWatchEventsClient.js";

// Set the parameters
export const params = {
  Name: "DEMO_EVENT",
  RoleArn: "IAM_ROLE_ARN", //IAM_ROLE_ARN
  ScheduleExpression: "rate(5 minutes)",
  State: "ENABLED",
};

export const run = async () => {
  try {
    const data = await cweClient.send(new PutRuleCommand(params));
    console.log("Success, scheduled rule created; Rule ARN:", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
// Uncomment this line to run execution within this file.
// run();
```

- Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK for JavaScript.
- Untuk rincian selengkapnya API, lihatPutRulediAWS SDK for JavaScriptReferensi API.

SDK for SDK for JavaScript V2

> **Tip**
>
> Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'REGION'});

// Create CloudWatchEvents service object
var cwevents = new AWS.CloudWatchEvents({apiVersion: '2015-10-07'});

var params = {
  Name: 'DEMO_EVENT',
  RoleArn: 'IAM_ROLE_ARN',
  ScheduleExpression: 'rate(5 minutes)',
  State: 'ENABLED'
};

cwevents.putRule(params, function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.RuleArn);
  }
});
```

- Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK for JavaScript.
- Untuk rincian selengkapnya API, lihatPutRulediAWS SDK for JavaScriptReferensi API.

Untuk daftar lengkapAWSPanduan pengembang SDK dan contoh kode, lihatMenggunakan CloudWatch Peristiwa denganAWSSDK (p. 99). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

mengirim CloudWatch Acara peristiwa menggunakanAWSSDK

Contoh kode berikut ini menunjukkan cara mengirim Amazon CloudWatch Peristiwa events.

Java

SDK for Java 2.x

Tip

Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

```
    public static void putCWEvents(CloudWatchEventsClient cwe, String resourceArn )
{
        try {

            final String EVENT_DETAILS =
                "{ \"key1\": \"value1\", \"key2\": \"value2\" }";

            PutEventsRequestEntry requestEntry = PutEventsRequestEntry.builder()
                .detail(EVENT_DETAILS)
                .detailType("sampleSubmitted")
                .resources(resourceArn)
                .source("aws-sdk-java-cloudwatch-example")
                .build();

            PutEventsRequest request = PutEventsRequest.builder()
                .entries(requestEntry)
                .build();

            cwe.putEvents(request);
            System.out.println("Successfully put CloudWatch event");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Untuk rincian selengkapnya API, lihatPutEventsdiAWS SDK for Java 2.xReferensi API.

JavaScript

SDK for SDK for JavaScript V3

Tip

Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

Buat klien dalam modul terpisah dan ekspor.

```
import { CloudWatchEventsClient } from "@aws-sdk/client-cloudwatch-events";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon CloudWatch service client object.
export const cweClient = new CloudWatchEventsClient({ region: REGION });
```

Impor modul SDK dan klien dan panggil API.

```
// Import required AWS SDK clients and commands for Node.js
import { PutEventsCommand } from "@aws-sdk/client-cloudwatch-events";
import { cweClient } from "./libs/cloudWatchEventsClient.js";

// Set the parameters
export const params = {
  Entries: [
    {
      Detail: '{ "key1": "value1", "key2": "value2" }',
      DetailType: "appRequestSubmitted",
      Resources: [
        "RESOURCE_ARN", //RESOURCE_ARN
      ],
      Source: "com.company.app",
    },
  ],
};

export const run = async () => {
  try {
    const data = await cweClient.send(new PutEventsCommand(params));
    console.log("Success, event sent; requestID:", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
// Uncomment this line to run execution within this file.
// run();
```

- Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK for JavaScript.
- Untuk rincian selengkapnya API, lihatPutEventsdiAWS SDK for JavaScriptReferensi API.

SDK for SDK for JavaScript V2

Tip

Untuk mempelajari cara menyiapkan dan menjalankan contoh ini, lihatGitHub.

```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'REGION'});

// Create CloudWatchEvents service object
var cwevents = new AWS.CloudWatchEvents({apiVersion: '2015-10-07'});

var params = {
```

```
  Entries: [
    {
      Detail: '{ \"key1\": \"value1\", \"key2\": \"value2\" }',
      DetailType: 'appRequestSubmitted',
      Resources: [
        'RESOURCE_ARN',
      ],
      Source: 'com.company.app'
    }
  ]
};

cwevents.putEvents(params, function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.Entries);
  }
});
```

- Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK for JavaScript.
- Untuk rincian selengkapnya API, lihatPutEventsdiAWS SDK for JavaScriptReferensi API.

Untuk daftar lengkapAWSPanduan pengembang SDK dan contoh kode, lihatMenggunakan CloudWatch Peristiwa denganAWSSDK (p. 99). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan untuk Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Untuk informasi keamanan CloudWatch Events, lihat Keamanan di Amazon EventBridge.

Menandai Amazon AndaCloudWatchAcara Sumber Daya

> **Note**
>
> Amazon EventBridge adalah cara terbaik untuk mengelola peristiwa Anda. CloudWatchPeristiwa danEventBridgeadalah layanan dan API dasar yang sama, namunEventBridgemenyediakan lebih banyak fitur. Perubahan yang Anda buat di salah satuCloudWatchatauEventBridgeakan muncul di setiap konsol. Untuk informasi selengkapnya, lihatAmazonEventBridge.

Tag merupakan label atribut kustom yang Anda atau AWS tetapkan ke sumber daya AWS. Setiap tanda memiliki dua bagian:

- Sebuah kunci tag (misalnya, `CostCenter`, `Environment`, atau `Project`). Kunci tag peka terhadap huruf besar dan kecil.
- Bidang opsional yang dikenal sebagai nilai tanda (misalnya, `111122223333` atau `Production`). Mengabaikan nilai tag sama saja dengan menggunakan string kosong. Seperti kunci tag, nilai tag peka huruf besar dan kecil.

Tag membantu Anda melakukan hal berikut:

- Identifikasi dan organisir sumber daya AWS Anda. Banyak layanan AWS yang mendukung penandaan, sehingga Anda dapat menetapkan tanda yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya tersebut terkait. Contohnya, Anda dapat menugaskan tag yang sama keCloudWatchAturan Peristiwa yang Anda tetapkan ke instans EC2.
- Telusuri biaya AWS Anda. Anda mengaktifkan tag ini pada AWS Billing and Cost Management dasbor.AWS menggunakan tag untuk mengategorikan biaya Anda lalu mengirimkan laporan alokasi biaya bulanan kepada Anda. Untuk informasi selengkapnya, lihat Gunakan Tag Alokasi Biaya dalam AWS BillingPanduan Pengguna.

Bagian berikut menyediakan informasi selengkapnya tentang tag untukCloudWatchPeristiwa.

Sumber Daya yang DidukungCloudWatchPeristiwa

Sumber daya berikut diCloudWatchAcara dukungan penandaan:

- Aturan

Untuk informasi tentang menambahkan dan mengelola tag, lihat Mengelola Tag (p. 109).

Mengelola Tag

Tag terdiri atas `Key` dan `Value` properti pada sumber daya. Anda dapat menggunakanCloudWatchkonsol,AWS CLI, atauCloudWatchAPI Peristiwa untuk menambahkan, mengedit, atau menghapus nilai untuk properti ini. Untuk informasi tentang bekerja dengan tanda, lihat hal berikut:

- TagResource,UntagResource, danListTagsForResourcediAmazonCloudWatchReferensi API Peristiwa
- tag-resource,untag-resource, danlist-tags-forsumber dayadiAmazonCloudWatchReferensi CLI
- Bekerja dengan Editor Tag dalam Panduan Pengguna Grup Sumber Daya

Kesepakatan Penamaan dan Penggunaan Tag

Kesepakatan penamaan dan penggunaan dasar berikut berlaku untuk penggunaan tag denganCloudWatchSumber acara:

- Setiap sumber daya dapat memiliki maksimum tanda 50.
- Untuk setiap sumber daya, setiap tanda kunci harus unik, dan setiap tanda kunci hanya dapat memiliki satu nilai.
- Panjang tanda kunci maksimum adalah 128 karakter Unicode dalam UTF-8.
- Panjang tanda nilai maksimum adalah 256 karakter Unicode dalam UTF-8.
- Karakter yang diperbolehkan adalah huruf, angka, spasi yang dapat ditampilkan di UTF-8, serta karakter berikut: . : + = @ _ / - (tanda hubung).
- Tanda Kunci dan nilai peka terhadap huruf besar dan kecil. Sebagai praktik terbaik, tentukan strategi untuk memanfaatkan tag dan secara konsisten menerapkan strategi tersebut di semua jenis sumber daya. Misalnya, putuskan apakah akan menggunakan `Costcenter`, `costcenter`, atau `CostCenter` dan menggunakan kesepakatan yang sama untuk semua tag. Hindari penggunaan tag yang serupa dengan perlakuan kasus yang tidak konsisten.
- Awalan `aws:` dilarang untuk tag karena diperuntukkan untuk penggunaan AWS. Anda tidak dapat menyunting atau menghapus kunci atau nilai tag dengan awalan ini. Tag dengan prefiks ini tidak memengaruhi tag per kuota sumber daya.

Mencatat Panggilan API Amazon CloudWatch Events dengan AWS CloudTrail

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Amazon CloudWatch Events terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang dilakukan oleh pengguna, peran, atau layanan AWS di CloudWatch Events. CloudTrail merekam panggilan API yang dibuat oleh atau atas nama akun AWS Anda. Panggilan yang direkam mencakup panggilan dari konsol dan panggilan kode CloudWatch ke operasi API CloudWatch Events. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan peristiwa CloudTrail ke bucket Amazon S3, termasuk peristiwa untuk CloudWatch Events. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru dalam konsol CloudTrail di Riwayat peristiwa. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke CloudWatch Events, alamat IP asal permintaan, siapa yang membuat permintaan, kapan dibuatnya, dan detail lainnya.

Untuk mempelajari CloudTrail selengkapnya, termasuk cara mengonfigurasi dan mengaktifkannya, lihat AWS CloudTrail Panduan Pengguna.

Topik

- Informasi CloudWatch Events di CloudTrail (p. 111)
- Contoh: CloudWatch Catatan File Log (p. 112)

Informasi CloudWatch Events di CloudTrail

CloudTrail diaktifkan pada akun AWS Anda saat Anda membuat akun tersebut. Saat aktivitas peristiwa yang didukung terjadi di CloudWatch Events, aktivitas tersebut dicatat di peristiwa CloudTrail bersama peristiwa layanan AWS lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat Melihat Peristiwa dengan Riwayat Peristiwa CloudTrail.

Untuk pencatatan peristiwa berkelanjutan di akun AWS Anda, termasuk peristiwa untuk CloudWatch Events, buatlah jejak. Jejak memungkinkan CloudTrail mengirim file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi layanan AWS lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di log CloudTrail. Untuk informasi selengkapnya, lihat yang berikut:

- Ikhtisar untuk Membuat Jejak

- Layanan yang Didukung dan Integrasi CloudTrail
- Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima Berkas Log CloudTrail dari Berbagai Wilayah dan Menerima Berkas Log CloudTrail dari Berbagai Akun

CloudWatch Events mendukung pencatatan tindakan berikut sebagai peristiwa dalam berkas log CloudTrail:

- DeleteRule
- DescribeEventBus
- DescribeRule
- DisableRule
- EnableRule
- ListRuleNamesByTarget
- ListRules
- listTargetsByRule
- PutPermission
- PutRule
- PutTargets
- RemoveTargets
- TestEventPattern

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.
- Apakah permintaan dibuat oleh layanan AWS lain.

Untuk informasi lebih lanjut, lihat Elemen userIdentity CloudTrail.

Contoh: CloudWatch Catatan File Log

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang telah Anda tentukan. File log CloudTrail berisi satu atau beberapa entri log. Peristuwa mewakili satu permintaan dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. File log CloudTrail bukan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

Entri berkas log CloudTrail berikut menunjukkan bahwa pengguna memanggil tindakan PutRule CloudWatch Events.

```
{
        "eventVersion":"1.03",
        "userIdentity":{
           "type":"Root",
           "principalId":"123456789012",
           "arn":"arn:aws:iam::123456789012:root",
```

```
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext":{
               "attributes":{
                  "mfaAuthenticated":"false",
                  "creationDate":"2015-11-17T23:56:15Z"
               }
            }
         },
         "eventTime":"2015-11-18T00:11:28Z",
         "eventSource":"events.amazonaws.com",
         "eventName":"PutRule",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"AWS Internal",
         "userAgent":"AWS CloudWatch Console",
         "requestParameters":{
            "description":"",
            "name":"cttest2",
            "state":"ENABLED",
            "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
            "scheduleExpression":""
         },
         "responseElements":{
            "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
         },
         "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
         "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
         "eventType":"AwsApiCall",
         "apiVersion":"2015-10-07",
         "recipientAccountId":"123456789012"
}
```

Kuota CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Untuk informasi tentang kuota layanan CloudWatch Events dan EventBridge, lihat Kuota Amazon EventBridge.

Untuk informasi selengkapnya, lihat berikut ini.

- Amazon EventBridge
- Service Quotas EventBridge
- Referensi API Amazon EventBridge

Pemecahan Masalah CloudWatch Events

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Anda dapat menggunakan langkah-langkah dalam bagian ini untuk memecahkan masalah CloudWatch Events.

Topik

- Aturan saya dipicu tapi fungsi Lambda saya tidak dipanggil (p. 115)
- Saya baru saja membuat atau memodifikasi aturan, tetapi tidak cocok dengan peristiwa pengujian (p. 116)
- Aturan saya tidak memicu sendiri pada waktu yang ditentukan dalam ScheduleExpression (p. 117)
- Aturan saya tidak memicu pada waktu yang saya harapkan (p. 117)
- Aturan saya cocok dengan panggilan API IAM tapi aturan saya tidak dipicu (p. 117)
- Aturan saya tidak bekerja karena IAM role yang terkait dengan aturan diabaikan ketika aturan dipicu (p. 118)
- Saya membuat aturan dengan EventPattern yang seharusnya cocok dengan sumber daya, tapi saya tidak melihat peristiwa yang cocok dengan aturan (p. 118)
- Pengiriman peristiwa saya ke target tertunda (p. 118)
- Beberapa kejadian tidak pernah dikirimkan ke target saya (p. 118)
- Aturan saya dipicu lebih dari sekali dalam menanggapi satu peristiwa. Jaminan apa yang ditawarkan CloudWatch Events untuk memicu aturan atau menyampaikan peristiwa ke target? (p. 119)
- Mencegah Loop Tak Terbatas (p. 119)
- Kejadian saya tidak dikirim ke antrean Amazon SQS target (p. 119)
- Aturan saya dipicu, tapi saya tidak melihat pesan yang diterbitkan ke topik Amazon SNS saya (p. 120)
- Topik Amazon SNS saya masih memiliki izin untuk CloudWatch Events bahkan setelah saya menghapus aturan yang terkait dengan topik Amazon SNS (p. 121)
- Kunci syarat IAM mana yang dapat saya gunakan dengan CloudWatch Events? (p. 121)
- Bagaimana saya bisa tahu saat aturan CloudWatch Events rusak? (p. 121)

Aturan saya dipicu tapi fungsi Lambda saya tidak dipanggil

Pastikan Anda memiliki izin yang tepat untuk fungsi Lambda Anda. Jalankan perintah berikut dengan menggunakan AWS CLI (ganti nama fungsi dengan fungsi Anda dan gunakan Wilayah AWS fungsi Anda):

```
aws lambda get-policy --function-name MyFunction --region us-east-1
```

Anda akan melihat output yang serupa dengan yang berikut:

```
{
    "Policy": "{\"Version\":\"2012-10-17\",
    \"Statement\":[
        {\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:events:us-
east-1:123456789012:rule/MyRule\"}},
        \"Action\":\"lambda:InvokeFunction\",
        \"Resource\":\"arn:aws:lambda:us-east-1:123456789012:function:MyFunction\",
        \"Effect\":\"Allow\",
        \"Principal\":{\"Service\":\"events.amazonaws.com\"},
        \"Sid\":\"MyId\"}
    ],
    \"Id\":\"default\"}"
}
```

Jika Anda melihat pesan berikut:

```
A client error (ResourceNotFoundException) occurred when calling the GetPolicy operation:
 The resource you requested does not exist.
```

Atau, Anda melihat output tetapi Anda tidak dapat menemukan events.amazonaws.com sebagai entitas terpercaya dalam kebijakan, jalankan perintah berikut:

```
aws lambda add-permission \
--function-name MyFunction \
--statement-id MyId \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
```

> **Note**
>
> Jika kebijakannya salah, Anda juga dapat mengedit aturan di konsol CloudWatch Events dengan menghapus lalu menambahkannya kembali ke aturan. Konsol CloudWatch Events kemudian menetapkan izin yang benar pada target.
> Jika Anda menggunakan alias atau versi Lambda tertentu, tambahkan parameter `--qualifier` di perintah `aws lambda get-policy` dan `aws lambda add-permission`.
>
> ```
> aws lambda add-permission \
> --function-name MyFunction \
> --statement-id MyId \
> --action 'lambda:InvokeFunction' \
> --principal events.amazonaws.com \
> --source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
> --qualifier alias or version
> ```

Alasan lain fungsi Lambda akan gagal untuk memicu adalah jika kebijakan yang Anda lihat ketika menjalankan `get-policy` berisi bidang `SourceAccount`. Pengaturan `SourceAccount` mencegah CloudWatch Events meminta fungsi.

Saya baru saja membuat atau memodifikasi aturan, tetapi tidak cocok dengan peristiwa pengujian

Saat Anda membuat perubahan aturan atau perubahan targetnya, kejadian yang masuk mungkin tidak langsung mulai atau berhenti cocok dengan aturan baru atau yang diperbarui. Tunggu sebentar hingga

perubahan diterapkan. Jika, setelah periode singkat ini, peristiwa masih tidak cocok, Anda juga dapat memeriksa metrik CloudWatch untuk aturan Anda seperti `TriggeredRules`, `Invocations`, dan `FailedInvocations` untuk debugging lebih lanjut. Untuk informasi selengkapnya tentang metrik ini, lihat Metrik dan Dimensi Amazon CloudWatch Events di Panduan Pengguna Amazon CloudWatch.

Jika aturan dipicu oleh sebuah peristiwa dari layanan AWS, Anda juga dapat menggunakan tindakan `TestEventPattern` untuk menguji pola peristiwa aturan Anda dengan peristiwa pengujian untuk memastikan pola peristiwa aturan Anda sudah diatur dengan benar. Untuk informasi selengkapnya, lihat TestEventPattern di Referensi API Amazon CloudWatch Events.

Aturan saya tidak memicu sendiri pada waktu yang ditentukan dalam ScheduleExpression

ScheduleExpressions ada dalam UTC. Pastikan Anda telah mengatur jadwal untuk aturan agar memicu sendiri di zona waktu UTC. Jika ScheduleExpression benar, ikuti langkah-langkah di bawah Saya baru saja membuat atau memodifikasi aturan, tetapi tidak cocok dengan peristiwa pengujian (p. 116).

Aturan saya tidak memicu pada waktu yang saya harapkan

CloudWatch Events tidak mendukung pengaturan waktu mulai yang tepat saat Anda membuat aturan untuk dijalankan setiap periode waktu. Hitung mundur untuk waktu aktif dimulai segera setelah Anda membuat aturan.

Anda dapat menggunakan ekspresi cron untuk memanggil target pada waktu tertentu. Misalnya, Anda dapat menggunakan ekspresi cron untuk membuat aturan yang dipicu setiap 4 jam tepat pada 0 menit. Di konsol CloudWatch, Anda akan menggunakan ekspresi cron `0 0/4 * * ? *`, dan dengan AWS CLI Anda akan menggunakan ekspresi cron `cron(0 0/4 * * ? *)`. Misalnya, untuk membuat aturan bernama TestRule yang dipicu setiap 4 jam menggunakan AWS CLI, Anda akan mengetik berikut ini pada prompt perintah:

```
aws events put-rule --name TestRule --schedule-expression 'cron(0 0/4 * * ? *)'
```

Anda dapat menggunakan ekspresi cron `0/5 * * * ? *` untuk memicu aturan setiap 5 menit. Misalnya:

```
aws events put-rule --name TestRule --schedule-expression 'cron(0/5 * * * ? *)'
```

CloudWatch Events tidak memberikan presisi tingkat kedua dalam ekspresi jadwal. Resolusi terbaik untuk penggunaan ekspresi cron adalah satu menit. Karena sifat terdistribusi CloudWatch Events dan layanan target, dapat terjadi penundaan beberapa detik antara waktu ketika aturan dijadwalkan dipicu dan waktu ketika layanan target menjalankan sumber daya target. Aturan terjadwal Anda akan dipicu di menit itu tetapi tidak tepat pada detik ke-0.

Aturan saya cocok dengan panggilan API IAM tapi aturan saya tidak dipicu

Layanan IAM hanya tersedia di Wilayah US East (N. Virginia), sehingga peristiwa Panggilan API AWS dari IAM hanya tersedia di wilayah tersebut. Untuk informasi selengkapnya, lihat CloudWatchContoh Peristiwa dari Layanan yang Didukung (p. 42).

Aturan saya tidak bekerja karena IAM role yang terkait dengan aturan diabaikan ketika aturan dipicu

IAM role untuk aturan hanya digunakan untuk peristiwa terkait dengan pengaliran Kinesis. Untuk fungsi Lambda atau topik Amazon SNS, Anda perlu memberikan Izin berbasis sumber daya.

Pastikan titik akhir AWS STS wilayah Anda diaktifkan. CloudWatch Events berbicara dengan titik akhir AWS STS wilayah saat mengambil IAM role yang Anda berikan. Untuk informasi selengkapnya, lihat Mengaktifkan dan Menonaktifkan AWS STS di Wilayah AWS di Panduan Pengguna IAM.

Saya membuat aturan dengan EventPattern yang seharusnya cocok dengan sumber daya, tapi saya tidak melihat peristiwa yang cocok dengan aturan

Sebagian besar layanan di AWS menganggap tanda titik dua (:) atau garis miring (/) sebagai karakter yang sama di Amazon Resource Names (ARNs). Namun, CloudWatch Events menggunakan pencarian yang sama persis di pola peristiwa dan aturan. Pastikan untuk menggunakan karakter ARN yang benar saat membuat pola peristiwa sehingga cocok dengan sintaks ARN di peristiwa yang dicocokkan.

Selain itu, tidak setiap peristiwa mempunyai field sumber daya yang diisi (seperti peristiwa panggilan API AWS dari CloudTrail).

Pengiriman peristiwa saya ke target tertunda

CloudWatch Events mencoba untuk mengirimkan peristiwa ke target hingga 24 jam, kecuali jika sumber daya target Anda dibatasi. Upaya pertama dilakukan segera setelah kejadian tiba di alur kejadian. Jika layanan target mengalami masalah, CloudWatch Events secara otomatis menjadwal ulang pengiriman lain. Jika 24 jam telah berlalu sejak datangnya peristiwa, tidak ada lagi upaya yang dijadwalkan dan metrik `FailedInvocations` diterbitkan di CloudWatch. Kami menyarankan Anda untuk membuat alarm CloudWatch di metrik `FailedInvocations`.

Beberapa kejadian tidak pernah dikirimkan ke target saya

Jika target aturan CloudWatch Events dibatasi untuk waktu yang lebih lama, CloudWatch Events mungkin tidak mencoba ulang pengiriman. Sebagai contoh, jika target tidak ditetapkan untuk menangani lalu lintas peristiwa yang masuk dan layanan target membatasi permintaan yang dibuat CloudWatch Events atas nama Anda, CloudWatch Events mungkin tidak mencoba ulang pengiriman.

Aturan saya dipicu lebih dari sekali dalam menanggapi satu peristiwa. Jaminan apa yang ditawarkan CloudWatch Events untuk memicu aturan atau menyampaikan peristiwa ke target?

Dalam kasus yang jarang terjadi, aturan yang sama dapat dipicu lebih dari sekali untuk satu peristiwa atau waktu yang dijadwalkan, atau target yang sama dapat diminta lebih dari sekali untuk sebuah aturan tertentu yang dipicu.

Mencegah Loop Tak Terbatas

Di CloudWatch Events, aturan yang dibuat mungkin menyebabkan loop tak terbatas, di mana aturan dijalankan berulang kali. Misalnya, aturan mungkin mendeteksi bahwa ACL telah berubah di bucket S3, dan memicu perangkat lunak untuk mengubahnya ke keadaan yang diinginkan. Jika aturan tidak ditulis dengan hati-hati, perubahan berikutnya pada ACL akan mengaktifkan kembali aturan, yang membuat loop tak terbatas.

Untuk mencegah hal ini, tulis aturan agar tindakan yang dipicu tidak mengaktifkan kembali aturan yang sama. Misalnya, aturan Anda hanya dapat berlaku jika keadaan ACL buruk, bukan setelah perubahan apa pun.

Loop tak terbatas dapat dengan cepat mengakibatkan biaya yang lebih tinggi dari yang diperkirakan. Kami menyarankan agar Anda menggunakan penganggaran, yang akan memberi tahu Anda saat biaya melebihi kuota yang ditentukan. Untuk informasi selengkapnya, lihat Mengelola Biaya Anda dengan Anggaran.

Kejadian saya tidak dikirim ke antrean Amazon SQS target

Antrean Amazon SQS dapat dienkripsi. Jika Anda membuat aturan dengan antrean Amazon SQS dienkripsi sebagai target, Anda harus memiliki bagian berikut yang disertakan dalam kebijakan kunci KMS agar peristiwa berhasil dikirim ke antrean yang terenkripsi .

```
{
                "Sid": "Allow CWE to use the key",
                "Effect": "Allow",
                "Principal": {
                                "Service": "events.amazonaws.com"
                },
                "Action": [
                                "kms:Decrypt",
                                "kms:GenerateDataKey"
                ],
                "Resource": "*"
}
```

Aturan saya dipicu, tapi saya tidak melihat pesan yang diterbitkan ke topik Amazon SNS saya

Pastikan Anda memiliki izin yang tepat ditetapkan untuk topik Amazon SNS Anda. Jalankan perintah berikut dengan menggunakan AWS CLI (ganti topik ARN dengan topik Anda dan gunakan Wilayah AWS topik Anda):

```
aws sns get-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic"
```

Anda akan melihat atribut kebijakan yang mirip dengan berikut ini:

```
"{\"Version\":\"2012-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:Receive\",
\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}},{\"Sid\":
\"Allow_Publish_Events\",
\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}]}"
```

Jika Anda melihat kebijakan yang mirip dengan berikut ini, Anda hanya memiliki kumpulan kebijakan default:

```
"{\"Version\":\"2008-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:Receive\",
\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}}]}"
```

Jika Anda tidak melihat `events.amazonaws.com` dengan Mempublikasikan izin dalam kebijakan Anda, gunakan AWS CLI untuk mengatur atribut kebijakan topik.

Salin kebijakan saat ini dan tambahkan pernyataan berikut ke daftar pernyataan:

```
{\"Sid\":\"Allow_Publish_Events\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}
```

Kebijakan baru akan terlihat seperti yang dijelaskan sebelumnya.

Tetapkan atribut topik dengan AWS CLI:

```
aws sns set-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic" --attribute-name Policy --attribute-value NEW_POLICY_STRING
```

> **Note**
>
> Jika kebijakannya salah, Anda juga dapat mengedit aturan di konsol CloudWatch Events dengan menghapus lalu menambahkannya kembali ke aturan. CloudWatch Events menetapkan izin yang benar pada target.

Topik Amazon SNS saya masih memiliki izin untuk CloudWatch Events bahkan setelah saya menghapus aturan yang terkait dengan topik Amazon SNS

Saat Anda membuat aturan dengan Amazon SNS sebagai target, CloudWatch Events menambahkan izin ke topik Amazon SNS atas nama Anda. Jika Anda menghapus aturan segera setelah Anda membuatnya, CloudWatch Events mungkin tidak menghapus izin dari topik Amazon SNS Anda. Jika ini terjadi, Anda dapat menghapus izin dari topik menggunakan perintah set-topic-attributes Amazon SNS.

Kunci syarat IAM mana yang dapat saya gunakan dengan CloudWatch Events?

CloudWatch Events mendukung kunci syarat seluruh AWS (lihat Kunci yang Tersedia di Panduan Pengguna IAM), ditambah kunci syarat khusus layanan berikut.

Bagaimana saya bisa tahu saat aturan CloudWatch Events rusak?

Anda dapat menggunakan alarm berikut untuk memberi tahu Anda ketika aturan CloudWatch Events rusak.

Membuat alarm untuk menginformasikan saat aturan rusak

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Pilih Buat Alarm. Di panel Metrik CloudWatch berdasarkan Kategori, pilih Metrik Kejadian.
3. Di daftar metrik, pilih FailedInvocations.

4. Di atas grafik, pilih Statistik, Sum.
5. Untuk Periode, pilih satu nilai, misalnya 5 menit. Pilih Selanjutnya.
6. Di bawah Ambang Batas Alarm, untuk Nama, masukkan nama yang unik untuk alarm, misalnya myFailedRules. Untuk Deskripsi, masukkan deskripsi alarm, misalnya Aturan tidak mengirimkan peristiwa ke target.
7. Untuk is, pilih >= dan 1. Untuk for, masukkan 10.
8. Di Tindakan, untuk Setiap kali alarm ini, pilih Status adalah ALARM.
9. Untuk Kirim notifikasi ke, pilih topik Amazon SNS yang sudah ada atau buat topik baru. Untuk membuat topik SNS baru, pilih Daftar baru. Masukkan nama untuk topik Amazon SNS baru, misalnya: myFailedRules.
10. Untuk Daftar email, ketik daftar alamat email yang dipisahkan dengan tanda koma untuk menerima pemberitahuan ketika alarm berubah ke status ALARM.
11. Pilih Buat Alarm.

Riwayat Dokumen

> **Note**
>
> Amazon EventBridge adalah cara pilihan untuk mengelola peristiwa Anda. CloudWatch Events dan EventBridge adalah layanan dan API dasar yang sama, tetapi EventBridge menyediakan lebih banyak fitur. Perubahan yang Anda buat di CloudWatch atau EventBridge akan muncul di setiap konsol. Untuk informasi selengkapnya, lihat Amazon EventBridge.

Tabel berikut menjelaskan perubahan penting dalam setiap terbitan Panduan Pengguna CloudWatch Events, yang dimulai pada Juni 2018. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

update-history-change	pembaruan-riwayat-deskripsi	pembaruan-riwayat-tanggal
Dukungan untuk penandaan (p. 123)	Sekarang Anda dapat menandai beberapa sumber daya CloudWatch Events. Untuk informasi selengkapnya, lihat Menandai Sumber Daya Amazon CloudWatch Events dalam Panduan Pengguna Amazon CloudWatch Events.	21 Maret 2019
Dukungan untuk titik akhir Amazon VPC (p. 123)	Sekarang Anda dapat membuat koneksi privat antara VPC dan CloudWatch Events Anda. Untuk informasi selengkapnya, lihat Menggunakan CloudWatch Events dengan VPC Endpoint Antarmuka dalam Panduan Pengguna Amazon CloudWatch Events.	28 Juni 2018

Tabel berikut menjelaskan perubahan penting terkait Panduan Pengguna Amazon CloudWatch Events.

Perubahan	Deskripsi	Tanggal Rilis
CodeBuild sebagai target	CodeBuild ditambahkan sebagai target untuk aturan peristiwa. Untuk informasi selengkapnya, lihat Tutorial: Jadwalkan Pembangunan Otomatis Menggunakan CodeBuild (p. 30).	13 Desember 2017
AWS Batch sebagai target	AWS Batch ditambahkan sebagai target untuk aturan Peristiwa. Untuk informasi selengkapnya, lihat Peristiwa AWS Batch.	8 September 2017
CodePipeline dan peristiwa AWS Glue	Dukungan ditambahkan untuk peristiwa dari CodePipeline dan AWS Glue. Untuk informasi selengkapnya, lihat CodePipelinePeristiwa (p. 45) dan Peristiwa AWS Glue (p. 57).	8 September 2017

Perubahan	Deskripsi	Tanggal Rilis
Peristiwa CodeBuild dan CodeCommit	Dukungan ditambahkan untuk peristiwa dari CodeBuild dan CodeCommit. Untuk informasi selengkapnya, lihat CodeBuildPeristiwa (p. 44).	3 Agustus 2017
Target tambahan yang didukung	CodePipeline dan Amazon Inspector dapat menjadi target peristiwa.	29 Juni 2017
Dukungan untuk mengirim dan menerima peristiwa antara akun AWS	Akun AWS dapat mengirim peristiwa ke akun AWS. Untuk informasi selengkapnya, lihat Mengirim dan Menerima Peristiwa Antara Akun AWS (p. 82).	29 Juni 2017
Target tambahan yang didukung	Sekarang Anda dapat mengatur duaAWSlayanan sebagai target untuk tindakan acara: Instans Amazon EC2 (melalui Run Command), dan mesin status Step Functions. Untuk informasi selengkapnya, lihat Memulai dengan Amazon CloudWatch Events (p. 6).	7 Maret 2017
Peristiwa Amazon EMR	Dukungan ditambahkan untuk peristiwa untuk Amazon EMR. Untuk informasi selengkapnya, lihat Peristiwa Amazon EMR (p. 48).	7 Maret 2017
Peristiwa AWS Health	Dukungan ditambahkan untuk peristiwa untuk AWS Health. Untuk informasi selengkapnya, lihat Peristiwa AWS Health (p. 62).	1 Desember 2016
Peristiwa Amazon Elastic Container Service	Dukungan ditambahkan untuk peristiwa untuk Amazon ECS. Untuk informasi selengkapnya, lihat Peristiwa Amazon Elastic Container Service (p. 48).	21 November 2016
Peristiwa AWS Trusted Advisor	Dukungan ditambahkan untuk peristiwa untuk Trusted Advisor. Untuk informasi selengkapnya, lihat Peristiwa AWS Trusted Advisor (p. 78).	Selasa, 18 Nopember 2016
Peristiwa Amazon Elastic Block Store	Dukungan ditambahkan untuk peristiwa untuk Amazon EBS. Untuk informasi selengkapnya, lihat Peristiwa Amazon EBS (p. 46).	14 November 2016
Peristiwa AWS CodeDeploy	Dukungan ditambahkan untuk peristiwa untuk CodeDeploy. Untuk informasi selengkapnya, lihat Peristiwa AWS CodeDeploy (p. 44).	9 September 2016
Peristiwa terjadwal dengan perincian 1 menit	Dukungan ditambahkan untuk peristiwa terjadwal dengan perincian 1 menit. Untuk informasi selengkapnya, lihat Ekspresi Cron (p. 33) dan Ekspresi Rate (p. 36).	19 April 2016
Antrean Amazon Simple Queue Service sebagai target	Dukungan ditambahkan untuk antrean Amazon SQS sebagai target. Untuk informasi selengkapnya, lihat Apa Itu Amazon CloudWatch Events? (p. 1).	30 Maret 2016
Peristiwa Auto Scaling	Dukungan ditambahkan untuk peristiwa untuk hook siklus hidup Auto Scaling. Untuk informasi selengkapnya, lihat Peristiwa Amazon EC2 Auto Scaling (p. 47).	24 Februari 2016

Perubahan	Deskripsi	Tanggal Rilis
Layanan baru	Rilis awal CloudWatch Events.	14 Januari 2016

Daftar istilah AWS

Untuk terminologi AWS terbaru, lihat AWS daftar istilah di AWS Referensi Umum.

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