aws

Panduan Pengguna

CloudWatch Log Amazon

CloudWatch Log Amazon: Panduan Pengguna

Table of Contents

Apa itu Amazon CloudWatch Logs?

Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan mengakses file log Anda dari instans Amazon Elastic Compute Cloud (Amazon EC2), Route 53 AWS CloudTrail, dan sumber lainnya.

CloudWatch Log memungkinkan Anda untuk memusatkan log dari semua sistem, aplikasi, dan AWS layanan yang Anda gunakan, dalam satu layanan yang sangat skalabel. Anda kemudian dapat dengan mudah melihatnya, mencari kode atau pola kesalahan tertentu, memfilternya berdasarkan bidang tertentu, atau mengarsipkannya dengan aman untuk analisis masa depan. CloudWatch Log memungkinkan Anda untuk melihat semua log Anda, terlepas dari sumbernya, sebagai aliran peristiwa tunggal dan konsisten yang diurutkan berdasarkan waktu.

CloudWatch Log juga mendukung kueri log Anda dengan bahasa kueri yang kuat, mengaudit dan menutupi data sensitif di log, dan menghasilkan metrik dari log menggunakan filter atau format log yang disematkan.

CloudWatch Log mendukung dua kelas log. Grup log di kelas CloudWatch log Standar Log mendukung semua fitur CloudWatch Log. Grup log di kelas CloudWatch log Akses Jarang Log dikenakan biaya konsumsi yang lebih rendah dan mendukung subset dari kemampuan kelas Standar. Untuk informasi selengkapnya, lihat Kelas log.

Fitur

- Dua kelas log untuk fleksibilitas — CloudWatch Log menawarkan dua kelas log sehingga Anda dapat memiliki opsi hemat biaya untuk log yang jarang Anda akses. Anda juga memiliki opsi fitur lengkap untuk log yang memerlukan pemantauan waktu nyata atau fitur lainnya. Untuk informasi selengkapnya, lihat Kelas log.
- Kueri data log Anda — Anda dapat menggunakan Wawasan CloudWatch Log untuk mencari dan menganalisis data log Anda secara interaktif. Anda dapat melakukan kueri untuk membantu Anda merespons masalah operasional secara lebih efisien dan efektif. CloudWatch Logs Insights mencakup bahasa kueri yang dibuat khusus dengan beberapa perintah sederhana namun kuat. Kami menyediakan kueri contoh, deskripsi perintah, penyelesaian otomatis kueri, dan penemuan bidang log untuk membantu Anda memulai. Contoh kueri disertakan untuk beberapa jenis log AWS layanan. Untuk memulai, lihat Menganalisis data log dengan Wawasan CloudWatch Log.
- Deteksi dan debug menggunakan Live Tail — Anda dapat menggunakan Live Tail untuk memecahkan masalah insiden dengan cepat dengan melihat daftar streaming peristiwa log baru

saat tertelan. Anda dapat melihat, memfilter, dan menyorot log yang dicerna dalam waktu dekat, membantu Anda mendeteksi dan menyelesaikan masalah dengan cepat. Anda dapat memfilter log berdasarkan istilah yang Anda tentukan, dan juga menyorot log yang berisi istilah tertentu untuk membantu Anda menemukan apa yang Anda cari dengan cepat. Untuk informasi selengkapnya, lihat Gunakan Live Tail untuk melihat log dalam waktu dekat.

- Memantau log dari instans Amazon EC2 — Anda dapat menggunakan CloudWatch Log untuk memantau aplikasi dan sistem menggunakan data log. Misalnya, CloudWatch Log dapat melacak jumlah kesalahan yang terjadi di log aplikasi Anda dan mengirimi Anda pemberitahuan setiap kali tingkat kesalahan melebihi ambang batas yang Anda tentukan. CloudWatch Log menggunakan data log Anda untuk pemantauan; jadi, tidak ada perubahan kode yang diperlukan. Misalnya, Anda dapat memantau log aplikasi untuk istilah literal tertentu (seperti "NullReferenceException“) atau menghitung jumlah kemunculan istilah literal pada posisi tertentu dalam data log (seperti kode status “404" dalam log akses Apache). Ketika istilah yang Anda cari ditemukan, CloudWatch Log melaporkan data ke CloudWatch metrik yang Anda tentukan. Data log dienkripsi saat transit dan saat diam. Untuk memulai, lihat Memulai dengan CloudWatch Log.
- Memantau peristiwa yang AWS CloudTrail dicatat - Anda dapat membuat alarm CloudWatch dan menerima pemberitahuan aktivitas API tertentu seperti yang ditangkap oleh CloudTrail dan menggunakan notifikasi untuk melakukan pemecahan masalah. Untuk memulai, lihat Mengirim CloudTrail Acara ke CloudWatch Log di Panduan AWS CloudTrail Pengguna.
- Audit dan tutupi data sensitif — Jika Anda memiliki data sensitif di log, Anda dapat membantu melindunginya dengan kebijakan perlindungan data. Kebijakan ini memungkinkan Anda mengaudit dan menutupi data sensitif. Jika Anda mengaktifkan perlindungan data, maka secara default, data sensitif yang cocok dengan pengidentifikasi data yang Anda pilih ditutupi. Untuk informasi selengkapnya, lihat Membantu melindungi data log sensitif dengan masking.
- Retensi log — Secara default, log disimpan tanpa batas waktu dan tidak pernah kedaluwarsa. Anda dapat menyesuaikan kebijakan retensi untuk setiap grup log, menjaga retensi yang tak terbatas, atau memilih periode retensi antara 10 tahun dan satu hari.
- Arsipkan data log - Anda dapat menggunakan CloudWatch Log untuk menyimpan data log Anda dalam penyimpanan yang sangat tahan lama. Agen CloudWatch Logs memudahkan untuk dengan cepat mengirim data log yang diputar dan tidak diputar dari host dan ke layanan log. Anda kemudian dapat mengakses data log mentah ketika dibutuhkannya.
- Kueri DNS Route 53 Log — Anda dapat menggunakan CloudWatch Log untuk mencatat informasi tentang kueri DNS yang diterima Route 53. Untuk informasi selengkapnya, lihat Mencatat Log Kueri DNS dalam Panduan Developer Amazon Route 53.

AWS Layanan terkait

Layanan berikut digunakan bersama dengan CloudWatch Log:

- AWS CloudTrailadalah layanan web yang memungkinkan Anda memantau panggilan yang dilakukan ke CloudWatch Logs API untuk akun Anda, termasuk panggilan yang dilakukan oleh AWS Management Console, AWS Command Line Interface (AWS CLI), dan layanan lainnya. Saat CloudTrail logging diaktifkan, CloudTrail menangkap panggilan API di akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Setiap berkas log dapat berisi satu atau lebih catatan, tergantung pada berapa banyak tindakan yang harus dilakukan untuk memenuhi permintaan. Untuk informasi lebih lanjut tentang AWS CloudTrail, lihat Apa itu AWS CloudTrail? dalam AWS CloudTrail User Guide. Untuk contoh jenis data yang CloudWatch menulis ke dalam file CloudTrail log, lihatLogging CloudWatch Logs API dan operasi konsol di AWS CloudTrail.
- AWS Identity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke AWS sumber daya dengan aman bagi pengguna Anda. Gunakan IAM untuk mengendalikan orang yang dapat menggunakan sumber daya AWS Anda (autentikasi) dan sumber daya apa yang dapat digunakan dengan cara apa (otorisasi). Untuk informasi selengkapnya, lihat Apa yang dimaksud dengan IAM? dalam Panduan Pengguna IAM.
- Amazon Kinesis Data Streams adalah layanan web yang dapat Anda gunakan untuk pengambilan dan agregasi data yang cepat dan berkesinambungan. 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 Apa yang dimaksud dengan Amazon Kinesis Data Streams? dalam Panduan Developer Amazon Kinesis Data Streams.
- AWS Lambda adalah layanan web yang dapat Anda gunakan untuk membangun aplikasi yang merespons informasi baru dengan cepat. Unggah kode aplikasi Anda sebagai fungsi Lambda dan Lambda menjalankan kode Anda di infrastruktur komputasi dengan ketersediaan tinggi dan melakukan semua administrasi sumber daya komputasi, termasuk pemeliharaan server dan sistem operasi, penyediaan kapasitas dan penskalaan otomatis, deployment patch keamanan, dan pemantauan dan pencatatan kode. Yang perlu Anda lakukan adalah menyediakan kode di salah satu bahasa yang didukung Lambda. Untuk informasi lebih lanjut, lihat Apa itu AWS Lambda? di Panduan AWS Lambda Pengembang.

Harga

Saat Anda mendaftar AWS, Anda dapat memulai dengan CloudWatch Log secara gratis menggunakan Tingkat AWS Gratis.

Tarif standar berlaku untuk log yang disimpan oleh layanan lain menggunakan CloudWatch Log (misalnya, log aliran VPC Amazon dan log Lambda).

Untuk informasi selengkapnya tentang harga, lihat CloudWatch Harga Amazon.

Untuk informasi selengkapnya tentang cara menganalisis biaya dan penggunaan untuk CloudWatch Log dan CloudWatch, dan untuk praktik terbaik tentang cara mengurangi biaya, lihat CloudWatch penagihan dan biaya.

Konsep Amazon CloudWatch Log

Terminologi dan konsep yang menjadi pusat pemahaman dan penggunaan CloudWatch Log Anda dijelaskan di bawah ini.

Kelas log

> CloudWatch Log menawarkan dua kelas grup log. Kelas log Standar adalah opsi berfitur lengkap untuk log yang memerlukan pemantauan waktu nyata atau log yang sering Anda akses. Kelas log Akses Jarang adalah opsi berbiaya lebih rendah untuk log yang Anda akses lebih jarang. Ini mendukung subset dari kemampuan kelas log Standar.

Log acara

> Log acara adalah catatan dari beberapa aktivitas yang direkam oleh aplikasi atau sumber daya yang dipantau. Catatan peristiwa CloudWatch log yang dipahami Log berisi dua properti: stempel waktu kapan peristiwa terjadi, dan pesan peristiwa mentah. Pesan kejadian harus dikodekan dalam UTF-8.

Pengaliran Log

> Pengaliran log adalah urutan log acara yang berbagi sumber yang sama. Lebih khusus lagi, pengaliran log umumnya dimaksudkan untuk mewakili urutan kejadian yang berasal dari instans aplikasi atau sumber daya yang dipantau. Sebagai contoh, pengaliran log dapat dikaitkan dengan log akses Apache pada host tertentu. Saat Anda tidak lagi membutuhkan aliran log, Anda dapat menghapusnya menggunakan delete-log-stream perintah aws logs.

Grup log

Grup log menentukan grup pengaliran log yang berbagi pengaturan kontrol retensi, pemantauan, dan akses yang sama. Setiap pengaliran log harus termasuk dalam satu grup log. Misalnya, jika Anda memiliki pengaliran log terpisah untuk log akses Apache dari setiap host, Anda dapat mengelompokkan pengaliran log tersebut ke dalam grup log tunggal yang disebut `MyWebsite.com/Apache/access_log`.

Tidak ada batas jumlah pengaliran log yang dapat tergabung dalam satu grup log.

Filter metrik

Anda dapat menggunakan filter metrik untuk mengekstrak pengamatan metrik dari peristiwa yang dicerna dan mengubahnya menjadi titik data dalam CloudWatch metrik. Filter metrik ditetapkan untuk grup log, dan semua filter yang ditetapkan ke grup log diterapkan ke pengaliran log mereka.

Pengaturan retensi

Pengaturan retensi dapat digunakan untuk menentukan berapa lama peristiwa log disimpan di CloudWatch Log. Log acara yang kedaluwarsa dapat dihapus secara otomatis. Sama seperti filter metrik, pengaturan retensi juga ditetapkan ke grup log, dan retensi yang ditetapkan ke grup log diterapkan ke pengaliran log mereka.

Penagihan dan biaya Amazon CloudWatch Logs

Untuk informasi terperinci tentang cara menganalisis biaya dan penggunaan untuk CloudWatch Log dan CloudWatch, dan untuk praktik terbaik tentang cara mengurangi biaya, lihat CloudWatch penagihan dan biaya.

Untuk informasi selengkapnya tentang harga, lihat CloudWatch Harga Amazon.

Saat Anda mendaftarAWS, Anda dapat memulai dengan CloudWatch Log secara gratis menggunakan Tingkat AWS Gratis.

Tarif standar berlaku untuk log yang disimpan oleh layanan lain menggunakan CloudWatch Log (misalnya, log aliran VPC Amazon dan log Lambda).

Kelas log

CloudWatch Log menawarkan dua kelas grup log:

- Kelas CloudWatch log Standar Log adalah opsi berfitur lengkap untuk log yang memerlukan pemantauan waktu nyata atau log yang sering Anda akses.
- Class CloudWatch log Logs Infrequent Access adalah kelas log baru yang dapat Anda gunakan untuk mengkonsolidasikan log Anda secara hemat biaya. Kelas log ini menawarkan subset kemampuan CloudWatch Log termasuk konsumsi terkelola, penyimpanan, analisis log lintas akun, dan enkripsi dengan harga konsumsi per GB yang lebih rendah. Kelas log Akses Jarang sangat ideal untuk kueri ad-hoc dan analisis after-the-fact forensik pada log yang jarang diakses.

> **Note**
>
> Untuk biaya, kelas log Akses Standar dan Jarang hanya berbeda dalam biaya konsumsi. Biaya penyimpanan dan biaya Wawasan CloudWatch Log sama di setiap kelas log.

Untuk informasi selengkapnya tentang harga CloudWatch Log, lihat CloudWatch Harga Amazon.

> **Important**
>
> Setelah grup log dibuat, kelas lognya tidak dapat diubah.

Fitur yang didukung

Tabel berikut mencantumkan fitur untuk setiap kelas log.

	Standar	Akses Jarang	
Tertelan dan penyimpanan log yang dikelola sepenuhnya	✓	✓	
Fitur lintas akun	✓	✓	
Enkripsi dengan AWS KMS	✓	✓	

	Standar	Akses Jarang	
CloudWatch Perintah kueri Log Insights	✓	✓ (Kebanyakan perintah—lihatPerintah yang didukung di kelas log.)	
CloudWatch Logs Insights menemukan bidang	✓		
Bantuan kueri bahasa alami	✓		
CloudWatch Deteksi Anomali Log	✓		
Bandingkan dengan rentang waktu sebelumnya	✓		
Filter berlangganan	✓		
Ekspor ke Amazon S3	✓		
GetLogEventsdan operasi FilterLogEventsAPI	✓	Tidak didukung. Gunakan Wawasan CloudWatch Log untuk melihat peristiwa log yang disimpan dalam grup log di kelas log Akses Jarang.	
Filter metrik	✓		
Kontainer Insights log ingestion	✓		

	Standar	Akses Jarang	
Konsumsi log Lambda Insights	✓		
Perlindungan data sensitif dengan masking	✓		
Format metrik tertanam	✓		

Memulai dengan CloudWatch Log

Untuk mengumpulkan log dari instans Amazon EC2 dan server lokal ke dalam CloudWatch Log, gunakan agen terpadu. CloudWatch Ini memungkinkan Anda untuk mengumpulkan log dan metrik lanjutan dengan satu agen. Ini menawarkan dukungan di seluruh sistem operasi, termasuk server yang menjalankan Windows Server. Agen ini juga memberikan performa yang lebih baik.

Jika Anda menggunakan CloudWatch agen terpadu untuk mengumpulkan CloudWatch metrik, ini memungkinkan pengumpulan metrik sistem tambahan, untuk visibilitas tamu. Ini juga mendukung pengumpulan metrik khusus menggunakan `StatsD` atau `collectd`.

Untuk informasi selengkapnya, lihat Menginstal CloudWatch Agen di Panduan CloudWatch Pengguna Amazon.

Agen CloudWatch Logs yang lebih lama, yang hanya mendukung kumpulan log dari server yang menjalankan Linux, tidak digunakan lagi dan tidak lagi didukung. Untuk informasi tentang migrasi dari agen CloudWatch Log lama ke agen terpadu, lihat Membuat file konfigurasi CloudWatch agen dengan wizard.

Daftar Isi

Prasyarat

Untuk menggunakan Amazon CloudWatch Logs, Anda memerlukan AWS akun. AWS Akun Anda memungkinkan Anda menggunakan layanan (misalnya, Amazon EC2) untuk menghasilkan log yang dapat Anda lihat di CloudWatch konsol, antarmuka berbasis web. Selain itu, Anda dapat menginstal dan mengkonfigurasi AWS Command Line Interface (AWS CLI).

Mendaftar untuk Akun AWS

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

Untuk mendaftar untuk Akun AWS

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

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

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

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Siapkan Antarmuka Baris Perintah

Anda dapat menggunakan AWS CLI untuk melakukan operasi CloudWatch Log.

Untuk informasi tentang cara menginstal dan mengkonfigurasi AWS CLI, lihat Menyiapkan dengan Antarmuka Baris AWS Perintah di Panduan AWS Command Line Interface Pengguna.

Gunakan CloudWatch agen terpadu untuk memulai dengan CloudWatch Log

Untuk informasi selengkapnya tentang penggunaan CloudWatch agen terpadu untuk memulai CloudWatch Log, lihat Mengumpulkan Metrik dan Log dari Instans Amazon EC2 dan Server Lokal dengan CloudWatch Agen di Panduan Pengguna Amazon. CloudWatch Anda menyelesaikan langkah-langkah yang tercantum dalam bagian ini untuk menginstal, mengonfigurasi, dan memulai agen. Jika Anda tidak menggunakan agen untuk juga mengumpulkan CloudWatch metrik, Anda dapat mengabaikan bagian apa pun yang merujuk ke metrik.

Jika saat ini Anda menggunakan agen CloudWatch Log lama dan ingin bermigrasi menggunakan agen terpadu baru, sebaiknya gunakan wizard yang disertakan dalam paket agen baru. Wizard ini dapat membaca file konfigurasi agen CloudWatch Log Anda saat ini dan mengatur CloudWatch agen untuk mengumpulkan log yang sama. Untuk informasi selengkapnya tentang wizard, lihat Membuat File Konfigurasi CloudWatch Agen dengan Wizard di Panduan CloudWatch Pengguna Amazon.

Gunakan CloudWatch agen sebelumnya untuk memulai dengan CloudWatch Log

> **Important**
>
> CloudWatch menyertakan CloudWatch agen terpadu yang dapat mengumpulkan log dan metrik dari instans EC2 dan server lokal. Agen logs-only yang lebih lama tidak digunakan lagi dan tidak lagi didukung.
> Untuk informasi tentang migrasi dari agen khusus log yang lebih lama ke agen terpadu, lihat Membuat file konfigurasi CloudWatch agen dengan wizard.
> Sisa bagian ini menjelaskan penggunaan agen CloudWatch Log lama untuk pelanggan yang masih menggunakannya.

Menggunakan agen CloudWatch Log, Anda dapat mempublikasikan data log dari instans Amazon EC2 yang menjalankan Linux atau Windows Server, dan peristiwa yang dicatat dari. AWS CloudTrail Sebaiknya gunakan agen CloudWatch terpadu untuk mempublikasikan data log Anda. Untuk informasi selengkapnya tentang agen baru, lihat Mengumpulkan Metrik dan Log dari Instans Amazon EC2 dan Server Lokal dengan CloudWatch Agen di Panduan Pengguna Amazon. CloudWatch

Daftar Isi

CloudWatch Prasyarat agen log

Agen CloudWatch Logs memerlukan Python versi 2.7, 3.0, atau 3.3, dan salah satu versi Linux berikut:

- Amazon Linux versi 2014.03.02 atau yang lebih baru. Amazon Linux 2 tidak didukung
- Ubuntu Server versi 12.04, 14.04, atau 16.04
- CentOS versi 6, 6.3, 6.4, 6.5, atau 7.0
- Red Hat Enterprise Linux (RHEL) versi 6.5 atau 7.0
- Debian 8.0

Mulai Cepat: Instal dan konfigurasikan agen CloudWatch Log pada instance EC2 Linux yang sedang berjalan

> **Important**
>
> Agen log lama tidak digunakan lagi. CloudWatch menyertakan agen terpadu yang dapat mengumpulkan log dan metrik dari instans EC2 dan server lokal. Untuk informasi selengkapnya, lihat Memulai dengan CloudWatch Log.

Untuk informasi tentang migrasi dari agen CloudWatch Log lama ke agen terpadu, lihat Membuat file konfigurasi CloudWatch agen dengan wizard.
Agen log yang lebih lama hanya mendukung Python versi 2.6 hingga 3.5. Selain itu, agen CloudWatch Log yang lebih lama tidak mendukung Layanan Metadata Instans Versi 2 (IMDSv2). Jika server Anda menggunakan IMDSv2, Anda harus menggunakan agen terpadu yang lebih baru, bukan agen Log yang lebih lama. CloudWatch
Sisa bagian ini menjelaskan penggunaan agen CloudWatch Log lama untuk pelanggan yang masih menggunakannya.

Tip

CloudWatch menyertakan agen terpadu baru yang dapat mengumpulkan log dan metrik dari instans EC2 dan server lokal. Jika Anda belum menggunakan agen CloudWatch Log yang lebih lama, kami sarankan Anda menggunakan agen terpadu CloudWatch yang lebih baru. Untuk informasi selengkapnya, lihat Memulai dengan CloudWatch Log.
Selain itu, agen lama tidak mendukung Layanan Metadata Instans Versi 2 (IMDSv2). Jika server Anda menggunakan IMDSv2, Anda harus menggunakan agen terpadu yang lebih baru, bukan agen Log yang lebih lama. CloudWatch
Sisa bagian ini menjelaskan penggunaan agen CloudWatch Log yang lebih tua.

Konfigurasikan agen CloudWatch Log yang lebih lama pada instans Linux EC2 yang sedang berjalan

Anda dapat menggunakan penginstal agen CloudWatch Log pada instans EC2 yang ada untuk menginstal dan mengonfigurasi agen CloudWatch Log. Setelah instalasi selesai, log secara otomatis mengalir dari instans ke pengaliran log yang Anda buat saat menginstal agen. Agen mengonfirmasi bahwa itu telah dimulai dan tetap berjalan sampai Anda menonaktifkannya.

Selain menggunakan agen, Anda juga dapat mempublikasikan data log menggunakan AWS CLI, CloudWatch Logs SDK, atau CloudWatch Logs API. Yang paling AWS CLI cocok untuk menerbitkan data di baris perintah atau melalui skrip. CloudWatch Logs SDK paling cocok untuk menerbitkan data log langsung dari aplikasi atau membuat aplikasi penerbitan log Anda sendiri.

Langkah 1: Konfigurasikan peran IAM atau pengguna Anda untuk CloudWatch Log

Agen CloudWatch Log mendukung peran dan pengguna IAM. Jika instans Anda sudah memiliki IAM role yang terkait dengannya, pastikan Anda menyertakan kebijakan IAM di bawah ini. Jika Anda belum memiliki IAM role yang ditetapkan ke instans, Anda dapat menggunakan kredensial IAM untuk langkah berikutnya atau Anda dapat menetapkan IAM role ke instans tersebut. Untuk informasi selengkapnya, lihat Melampirkan IAM Role ke Instans.

Untuk mengonfigurasi peran IAM atau pengguna Anda untuk CloudWatch Log

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran.
3. Pilih peran dengan memilih nama peran (jangan mencentang kotak di samping nama).
4. Pilih Attach Policies (Lampirkan Kebijakan), Create Policy (Buat Kebijakan).

   Tab atau jendela peramban baru akan terbuka.
5. Pilih tab JSON dan ketik dokumen kebijakan JSON berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "*"
    ]
  }
 ]
}
```

6. Setelah Anda selesai, pilih Tinjau kebijakan. Validator Kebijakan melaporkan kesalahan sintaksis.
7. Di halaman Review Policy (Tinjau Kebijakan), ketikkan Name (Nama) dan Description (Deskripsi) (optional) untuk kebijakan yang sedang Anda buat. Tinjau Summary (Ringkasan) kebijakan

untuk melihat izin yang diberikan oleh kebijakan Anda. Kemudian pilih Buat kebijakan untuk menyimpan pekerjaan Anda.

8. Tutup tab atau jendela peramban, dan kembali ke halaman Add permissions (Tambahkan izin) untuk peran Anda. Pilih Refresh (Segarkan), lalu pilih kebijakan baru untuk dilampirkan ke peran Anda.
9. Pilih Lampirkan Kebijakan.

Langkah 2: Instal dan konfigurasikan CloudWatch Log pada instans Amazon EC2 yang ada

Proses untuk menginstal agen CloudWatch Log berbeda tergantung pada apakah instans Amazon EC2 Anda menjalankan Amazon Linux, Ubuntu, CentOS, atau Red Hat. Gunakan langkah-langkah yang sesuai untuk versi Linux di instans Anda.

Untuk menginstal dan mengonfigurasi CloudWatch Log pada instans Amazon Linux yang ada

Dimulai dengan Amazon Linux AMI 2014.09, agen CloudWatch Logs tersedia sebagai instalasi RPM dengan paket awslogs. Versi sebelumnya dari Amazon Linux dapat mengakses paket awslogs dengan memperbarui instans dengan perintah `sudo yum update -y`. Dengan menginstal paket awslogs sebagai RPM alih-alih menggunakan penginstal CloudWatch Log, instans Anda menerima pembaruan dan tambalan paket reguler AWS tanpa harus menginstal ulang agen Log secara manual. CloudWatch

> ⚠ Warning
>
> Jangan memperbarui agen CloudWatch Log menggunakan metode instalasi RPM jika sebelumnya Anda menggunakan skrip Python untuk menginstal agen. Melakukannya dapat menyebabkan masalah konfigurasi yang mencegah agen CloudWatch Log mengirim log Anda CloudWatch.

1. Hubungkan ke instans Amazon Linux Anda. Untuk informasi selengkapnya, lihat Connect to Your Instance di Panduan Pengguna Amazon EC2.

   Untuk informasi selengkapnya tentang masalah koneksi, lihat Pemecahan Masalah Menghubungkan ke Instans Anda di Panduan Pengguna Amazon EC2.
2. Perbarui instans Amazon Linux Anda untuk mengambil perubahan terbaru dalam repositori paket.

```
sudo yum update -y
```

3. Instal`awslogs` paket. Ini adalah metode yang direkomendasikan untuk menginstal awslogs di instans Amazon Linux.

```
sudo yum install -y awslogs
```

4. Edit file `/etc/awslogs/awslogs.conf` untuk mengonfigurasi log yang akan dilacak. Untuk informasi selengkapnya tentang mengedit file ini, lihat CloudWatch Referensi agen log.
5. Secara default, `/etc/awslogs/awscli.conf` menunjuk ke Wilayah us-east-1. Untuk mendorong log Anda ke Wilayah yang berbeda, edit file `awscli.conf` dan tentukan Wilayah tersebut.
6. Mulai layanan `awslogs`.

```
sudo service awslogs start
```

   Jika Anda menjalankan Amazon Linux 2, mulai layanan `awslogs` dengan perintah berikut.

```
sudo systemctl start awslogsd
```

7. (Opsional) Periksa file `/var/log/awslogs.log` untuk kesalahan yang dicatat saat memulai layanan.
8. (Opsional) Jalankan perintah berikut untuk memulai layanan `awslogs` pada setiap boot sistem.

```
sudo chkconfig awslogs on
```

   Jika Anda menjalankan Amazon Linux 2, gunakan perintah berikut untuk memulai layanan pada setiap boot sistem.

```
sudo systemctl enable awslogsd.service
```

9. Anda akan melihat grup log dan aliran log yang baru dibuat di CloudWatch konsol setelah agen berjalan selama beberapa saat.

   Untuk informasi selengkapnya, lihat Lihat data log yang dikirim ke CloudWatch Log.

Untuk menginstal dan mengkonfigurasi CloudWatch Log pada Server Ubuntu, CentOS, atau contoh Red Hat yang ada

Jika Anda menggunakan AMI yang menjalankan Ubuntu Server, CentOS, atau Red Hat, gunakan prosedur berikut untuk menginstal agen CloudWatch Log secara manual pada instance Anda.

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat Connect to Your Instance di Panduan Pengguna Amazon EC2.

   Untuk informasi selengkapnya tentang masalah koneksi, lihat Pemecahan Masalah Menghubungkan ke Instans Anda di Panduan Pengguna Amazon EC2.

2. Jalankan penginstal agen CloudWatch Log menggunakan salah satu dari dua opsi. Anda dapat menjalankannya langsung dari internet, atau mengunduh file dan menjalankannya secara mandiri.

   > **Note**
   >
   > Jika Anda menjalankan CentOS 6.x, Red Hat 6.x, atau Ubuntu 12.04, gunakan langkah-langkah untuk mengunduh dan menjalankan penginstal mandiri. Menginstal agen CloudWatch Log langsung dari internet tidak didukung pada sistem ini.

   > **Note**
   >
   > Di Ubuntu, jalankan `apt-get update` sebelum menjalankan perintah di bawah ini.

   Untuk menjalankannya secara langsung dari internet, gunakan arahan berikut dan ikuti arahannya:

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
   ```

   ```
   sudo python ./awslogs-agent-setup.py --region us-east-1
   ```

   Jika perintah sebelumnya tidak berfungsi, coba hal berikut:

```
sudo python3 ./awslogs-agent-setup.py --region us-east-1
```

Untuk mengunduh dan menjalankannya secara mandiri, gunakan perintah berikut dan ikuti arahannya:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py -O
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/
AgentDependencies.tar.gz -O
```

```
tar xvf AgentDependencies.tar.gz -C /tmp/
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/
AgentDependencies
```

Anda dapat menginstal agen CloudWatch Log dengan menentukan us-east-1, us-west-1, us-west-2, ap-south-1, ap-south-1, ap-northeast-2, ap-southeast-2, ap-southeast-1, ap-southeast-1, ap-southeast-2, ap-northeast-1 Wilayah eu-central-1, eu-west-1, atau sa-east-1, atau sa-east-1.

> **Note**
>
> Untuk informasi selengkapnya tentang versi saat ini dan riwayat versi `awslogs-agent-setup`, lihat CHANGELOG.txt.

Installer agen CloudWatch Log memerlukan informasi tertentu selama penyiapan. Sebelum memulai, Anda perlu mengetahui berkas log mana yang akan dipantau dan format stempel waktunya. Anda juga harus menyiapkan informasi berikut.

Item	Deskripsi
AWS ID kunci akses	Tekan Enter jika menggunakan IAM role. Jika tidak, masukkan ID kunci AWS akses Anda.

Item	Deskripsi
AWS kunci akses rahasia	Tekan Enter jika menggunakan IAM role. Jika tidak, masukkan kunci akses AWS rahasia Anda.
Nama Wilayah default	Tekan Enter. Default-nya adalah us-east-2. Anda dapat mengatur ini menjadi us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, atau sa-east-1.
Format output default	Kosongkan dan tekan Enter.
Jalur berkas log untuk diunggah	Lokasi file yang berisi data log untuk dikirim. Penginstal akan menyarankan jalur untuk Anda.
Nama Grup Log tujuan	Nama untuk grup log Anda. Penginstal akan menyarankan nama grup log untuk Anda.
Nama Pengaliran Log tujuan	Secara default, ini adalah nama host. Penginstal akan menyarankan nama host untuk Anda.
Format stempel waktu	Tentukan format stempel waktu dalam berkas log yang ditentukan. Pilih custom (khusus) untuk menentukan format Anda sendiri.
Posisi awal	Cara data diunggah. Atur ini menjadi start_of_file untuk mengunggah segala sesuatu dalam file data. Atur menjadi end_of_file untuk mengunggah hanya data yang baru ditambahkan.

Setelah menyelesaikan langkah-langkah ini, penginstal menanyakan tentang konfigurasi berkas log lainnya. Anda dapat menjalankan proses sebanyak yang Anda inginkan untuk setiap berkas log. Jika Anda tidak memiliki berkas log lagi untuk dipantau, pilih N saat diminta oleh penginstal untuk menyiapkan log lain. Untuk informasi selengkapnya tentang pengaturan di file konfigurasi agen, lihat CloudWatch Referensi agen log.

> **Note**
>
> Mengonfigurasi beberapa sumber log untuk mengirim data ke satu pengaliran log tidaklah didukung.

3. Anda akan melihat grup log dan aliran log yang baru dibuat di CloudWatch konsol setelah agen berjalan selama beberapa saat.

   Untuk informasi selengkapnya, lihat Lihat data log yang dikirim ke CloudWatch Log.

Mulai Cepat: Instal dan konfigurasikan agen CloudWatch Log pada instans Linux EC2 saat diluncurkan

> **Tip**
>
> Agen CloudWatch Log lama yang dibahas di bagian ini sedang menuju penghentian. Kami sangat menyarankan agar Anda menggunakan CloudWatch agen terpadu baru yang dapat mengumpulkan log dan metrik. Selain itu, agen CloudWatch Logs yang lebih lama memerlukan Python 3.3 atau yang lebih lama, dan versi ini tidak diinstal pada instans EC2 baru secara default. Untuk informasi selengkapnya tentang CloudWatch agen terpadu, lihat Menginstal CloudWatch Agen.
> Sisa bagian ini menjelaskan penggunaan agen CloudWatch Log yang lebih tua.

Menginstal agen CloudWatch Logs yang lebih lama pada instans Linux EC2 saat diluncurkan

Anda dapat menggunakan data pengguna Amazon EC2, fitur Amazon EC2 yang memungkinkan informasi parametrik diteruskan ke instans saat diluncurkan, untuk menginstal dan mengonfigurasi agen Log pada instance CloudWatch tersebut. Untuk meneruskan informasi instalasi dan konfigurasi agen CloudWatch Log ke Amazon EC2, Anda dapat menyediakan file konfigurasi di lokasi jaringan seperti bucket Amazon S3.

Mengonfigurasi beberapa sumber log untuk mengirim data ke satu pengaliran log tidaklah didukung.

Prasyarat

Buat file konfigurasi agen yang menjelaskan semua grup log dan pengaliran log Anda. Ini adalah file teks yang menjelaskan berkas log yang akan dipantau serta grup log dan pengaliran log untuk mengunggah berkas log. Agen mengonsumsi file konfigurasi ini dan mulai memantau dan mengunggah semua berkas log yang dijelaskan di dalamnya. Untuk informasi selengkapnya tentang pengaturan di file konfigurasi agen, lihat CloudWatch Referensi agen log.

Berikut ini adalah sampel dari file konfigurasi agen untuk Amazon Linux 2

```
[general]
state_file = /var/lib/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Berikut ini adalah sampel dari file konfigurasi agen untuk Ubuntu

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Untuk mengonfigurasi IAM role

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Policies (Kebijakan), Create Policy (Buat Kebijakan).
3. Di halaman Create Policy (Buat Kebijakan), untuk Create Your Own Policy (Buat Kebijakan Anda Sendiri), pilih Select (Pilih). Untuk informasi selengkapnya tentang membuat kebijakan khusus, lihat Kebijakan IAM untuk Amazon EC2 di Panduan Pengguna Amazon EC2.
4. Di halaman Review Policy (Tinjau Kebijakan), untuk Policy Name (Nama Kebijakan), ketikkan nama untuk kebijakan tersebut.
5. Untuk Policy Document (Dokumen Kebijakan), tempelkan kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
```

```
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::myawsbucket/*"
            ]
        }
    ]
}
```

6. Pilih Buat Kebijakan.
7. Di panel navigasi, pilih Roles (Peran), Create New Role (Buat Peran Baru).
8. Di halaman Set Role Name (Tetapkan Nama Peran), ketik nama untuk peran tersebut, lalu pilih Next Step (Langkah Selanjutnya).
9. Di halaman Select Role Type (Pilih Jenis Peran), pilih Select (Pilihan) di samping Amazon EC2.
10. Di halaman Attach Policy (Lampirkan Kebijakan), di header tabel, pilih Policy Type (Jenis Kebijakan), Customer Managed (Dikelola Pelanggan).
11. Pilih kebijakan IAM yang sudah Anda buat, lalu pilih Next Step (Langkah Selanjutnya).
12. Pilih Buat peran.

    Untuk informasi selengkapnya tentang pengguna dan kebijakan, lihat Pengguna dan Grup IAM dan Mengelola Kebijakan IAM di Panduan Pengguna IAM.

Untuk meluncurkan instance baru dan mengaktifkan CloudWatch Log

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pilih Luncurkan Instans.

   Untuk informasi selengkapnya, lihat Meluncurkan Instans di Panduan Pengguna Amazon EC2.

3. Di halaman Langkah 1: Pilih Amazon Machine Image (AMI) pilih tipe instans Linux yang akan diluncurkan, lalu di halaman Langkah 2: Pilih Tipe Instans, pilih Selanjutnya: Konfigurasi Detail Instans.

   Pastikan bahwa cloud-init termasuk dalam Amazon Machine Image (AMI) Anda. Amazon Linux AMI, dan AMI untuk Ubuntu dan RHEL sudah menyertakan cloud-init, tetapi CentOS dan AMI lainnya mungkin tidak. AWS Marketplace
4. Di halaman Langkah 3: Konfigurasi Detail Instans, untuk IAM role, pilih IAM role yang sudah Anda buat.
5. Di Advanced Details (Detail Lanjutan), untuk User data (Data pengguna), tempelkan skrip berikut ke dalam kotak. Kemudian perbarui skrip tersebut dengan mengubah nilai opsi -c menjadi lokasi file konfigurasi agen Anda:

   ```
   #!/bin/bash
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
   setup.py -O
   chmod +x ./awslogs-agent-setup.py
   ./awslogs-agent-setup.py -n -r us-east-1 -c s3://DOC-EXAMPLE-BUCKET1/my-config-file
   ```
6. Buat perubahan lain pada instans, tinjau pengaturan peluncuran Anda, lalu pilih Launch (Luncurkan).
7. Anda akan melihat grup log dan aliran log yang baru dibuat di CloudWatch konsol setelah agen berjalan selama beberapa saat.

   Untuk informasi selengkapnya, lihat Lihat data log yang dikirim ke CloudWatch Log.

Mulai Cepat: Aktifkan instans Amazon EC2 Anda yang menjalankan Windows Server 2016 untuk mengirim log ke Log menggunakan agen CloudWatch Log CloudWatch

> **Tip**
>
> CloudWatch menyertakan agen terpadu baru yang dapat mengumpulkan log dan metrik dari instans EC2 dan server lokal. Kami menyarankan Anda menggunakan agen terpadu CloudWatch yang lebih baru. Untuk informasi selengkapnya, lihat Memulai dengan CloudWatch Log.

Sisa bagian ini menjelaskan penggunaan agen CloudWatch Log yang lebih tua.

Aktifkan instans Amazon EC2 Anda yang menjalankan Windows Server 2016 untuk mengirim log ke Log menggunakan agen CloudWatch Log yang lebih lama CloudWatch

Ada beberapa metode yang dapat Anda gunakan untuk mengaktifkan instance yang menjalankan Windows Server 2016 untuk mengirim CloudWatch log ke Log. Langkah-langkah di bagian ini menggunakan Systems Manager Run Command. Untuk informasi tentang metode lain yang mungkin, lihat Mengirim Log, Peristiwa, dan Penghitung Kinerja ke Amazon CloudWatch.

Langkah-langkah

- Unduh file konfigurasi contoh
- Konfigurasikan file JSON untuk CloudWatch
- Membuat peran IAM untuk Systems Manager
- Memverifikasi prasyarat Systems Manager
- Memverifikasi Akses Internet
- Aktifkan CloudWatch Log menggunakan Systems Manager Run Command

Unduh file konfigurasi contoh

Unduh file contoh berikut ke komputer Anda: `AWS.EC2.Windows.CloudWatch.json`.

Konfigurasikan file JSON untuk CloudWatch

Anda menentukan log mana yang akan dikirim CloudWatch dengan menentukan pilihan Anda dalam file konfigurasi. Proses untuk membuat file ini dan menentukan pilihan Anda dapat memakan waktu 30 menit atau lebih untuk diselesaikan. Setelah Anda menyelesaikan tugas ini satu kali, Anda dapat menggunakan kembali file konfigurasi di semua instans Anda.

Langkah-langkah

- Langkah 1: Aktifkan CloudWatch Log
- Langkah 2: Konfigurasikan pengaturan untuk CloudWatch
- Langkah 3: Konfigurasi data untuk mengirim
- Langkah 4: Konfigurasi kontrol aliran

- Langkah 5: Simpan konten JSON

Langkah 1: Aktifkan CloudWatch Log

Di bagian atas file JSON, ubah "false" menjadi "true" untuk `IsEnabled`:

```
"IsEnabled": true,
```

Langkah 2: Konfigurasikan pengaturan untuk CloudWatch

Tentukan kredensial, Wilayah, nama grup log, dan namespace pengaliran log. Hal ini memungkinkan instance untuk mengirim data log ke CloudWatch Log. Untuk mengirim data log yang sama ke lokasi yang berbeda, Anda dapat menambahkan bagian tambahan dengan ID unik (misalnya, "CloudWatchLogs2" dan CloudWatchLogs 3") dan Wilayah yang berbeda untuk setiap ID.

Untuk mengkonfigurasi pengaturan untuk mengirim data log ke CloudWatch Log

1. Dalam file JSON, temukan bagian `CloudWatchLogs`.

   ```
   {
       "Id": "CloudWatchLogs",
       "FullName":
    "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
       "Parameters": {
           "AccessKey": "",
           "SecretKey": "",
           "Region": "us-east-1",
           "LogGroup": "Default-Log-Group",
           "LogStream": "{instance_id}"
       }
   },
   ```

2. Biarkan bidang `AccessKey` dan `SecretKey` tetap kosong. Anda mengonfigurasi kredensial menggunakan IAM role.
3. Untuk `Region`, ketik Wilayah untuk mengirim data log (misalnya, `us-east-2`).
4. Untuk `LogGroup`, ketik nama untuk grup log Anda. Nama ini muncul di layar Grup Log di CloudWatch konsol.
5. Untuk `LogStream`, ketik pengaliran log tujuan. Nama ini muncul di layar Grup Log > Streams di CloudWatch konsol.

Jika Anda menggunakan `{instance_id}`, yaitu default-nya, nama pengaliran log adalah ID instans dari instans ini.

Jika Anda menentukan nama aliran log yang belum ada, CloudWatch Log secara otomatis membuatnya untuk Anda. Anda dapat menentukan nama pengaliran log menggunakan string literal, variabel yang telah ditetapkan `{instance_id}`, `{hostname}`, dan `{ip_address}`, atau kombinasinya.

Langkah 3: Konfigurasi data untuk mengirim

Anda dapat mengirim data log peristiwa, data Event Tracing for Windows (ETW), dan data log lainnya ke CloudWatch Log.

Untuk mengirim data log peristiwa aplikasi Windows ke CloudWatch Log

1. Dalam file JSON, temukan bagian `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.
   - **4** - Unggah hanya pesan informasi.

   Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim data log keamanan ke CloudWatch Log

1. Dalam file JSON, temukan bagian `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Untuk `Levels`, ketik **7** untuk mengunggah semua pesan.

Untuk mengirim data log peristiwa sistem ke CloudWatch Log

1. Dalam file JSON, temukan bagian `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.
   - **4** - Unggah hanya pesan informasi.

   Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim jenis data log peristiwa lainnya ke CloudWatch Log

1. Dalam file JSON, tambahkan bagian baru. Setiap bagian harus memiliki `Id` yang unik.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Untuk `Id`, ketik nama untuk log yang akan diunggah (misalnya, **`WindowsBackup`**).
3. Untuk `LogName`, ketik nama log yang akan diunggah. Anda dapat menemukan nama log sebagai berikut.

   a. Buka Event Viewer.

   b. Di panel navigasi, pilih Applications and Services Logs (Log Aplikasi dan Layanan).

   c. Buka log, lalu pilih Actions (Tindakan), Properties (Properti).
4. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.
   - **4** - Unggah hanya pesan informasi.

   Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim Event Tracing untuk data Windows ke CloudWatch Log

ETW (Event Tracing for Windows) menyediakan mekanisme pencatatan log yang efisien dan terperinci yang dapat digunakan aplikasi untuk menuliskan log. Setiap ETW dikendalikan oleh manajer sesi yang dapat memulai dan menghentikan sesi pencatatan. Setiap sesi memiliki penyedia dan satu atau beberapa konsumen.

1. Dalam file JSON, temukan bagian `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Untuk `LogName`, ketik nama log yang akan diunggah.
3. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.
   - **4** - Unggah hanya pesan informasi.

   Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim log khusus (file log berbasis teks apa pun) ke Log CloudWatch

1. Dalam file JSON, temukan bagian `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
```

```
    }
},
```

2. Untuk `LogDirectoryPath`, ketik jalur tempat log disimpan di instans Anda.
3. Untuk `TimestampFormat`, ketik format stempel waktu yang akan digunakan. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik Custom Date and Time Format Strings di MSDN.

> **Important**
>
> Berkas log sumber Anda harus memiliki stempel waktu di awal setiap baris log dan harus ada spasi setelah stempel waktu.

4. Untuk `Encoding`, ketik pengodean file yang akan digunakan (misalnya, UTF-8). Untuk daftar nilai yang didukung, lihat topik Encoding Class di MSDN.

> **Note**
>
> Gunakan nama pengodean, bukan nama tampilan.

5. (Opsional) Untuk `Filter`, ketik prefiks nama log. Biarkan parameter ini kosong untuk memantau semua file. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik FileSystemWatcherFilter Properti di MSDN.
6. (Opsional) Untuk `CultureName`, ketik lokal tempat stempel waktu dicatat. Jika `CultureName` kosong, default-nya adalah lokal yang sama yang saat ini digunakan oleh instans Windows Anda. Untuk informasi selengkapnya, lihat kolom `Language tag` di tabel dalam topik Product Behavior di MSDN.

> **Note**
>
> Nilai `div`, `div-MV`, `hu`, dan `hu-HU` tidak didukung.

7. (Opsional) Untuk `TimeZoneKind`, ketik `Local` atau `UTC`. Anda dapat mengatur ini untuk memberikan informasi zona waktu ketika tidak ada informasi zona waktu yang disertakan dalam stempel waktu log Anda. Jika parameter ini dibiarkan kosong dan jika stempel waktu Anda tidak menyertakan informasi zona waktu, CloudWatch Log default ke zona waktu lokal. Parameter ini diabaikan jika stempel waktu Anda sudah berisi informasi zona waktu.

8. (Opsional) Untuk `LineCount`, ketik jumlah baris di header untuk mengidentifikasi berkas log. Sebagai contoh, berkas log IIS memiliki header yang hampir identik. Anda bisa memasukkan **5**, yang akan membaca tiga baris pertama header berkas log untuk mengidentifikasinya. Dalam berkas log IIS, baris ketiga adalah tanggal dan stempel waktu, tetapi stempel waktu tidak selalu dijamin akan berbeda antara berkas log. Untuk alasan ini, sebaiknya sertakan setidaknya satu baris data log aktual untuk membuat sidik jari berkas log secara unik.

Untuk mengirim data log IIS ke CloudWatch Log

1. Dalam file JSON, temukan bagian `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Untuk `LogDirectoryPath`, ketik folder tempat IIS log disimpan untuk situs individual (misalnya, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> Hanya format log W3C yang didukung. Format IIS, NCSA, dan Custom tidak didukung.

3. Untuk `TimestampFormat`, ketik format stempel waktu yang akan digunakan. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik Custom Date and Time Format Strings di MSDN.
4. Untuk `Encoding`, ketik pengodean file yang akan digunakan (misalnya, UTF-8). Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik Encoding Class di MSDN.

> **Note**
>
> Gunakan nama pengodean, bukan nama tampilan.

5. (Opsional) Untuk `Filter`, ketik prefiks nama log. Biarkan parameter ini kosong untuk memantau semua file. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik FileSystemWatcherFilter Properti di MSDN.
6. (Opsional) Untuk `CultureName`, ketik lokal tempat stempel waktu dicatat. Jika `CultureName` kosong, default-nya adalah lokal yang sama yang saat ini digunakan oleh instans Windows Anda. Untuk informasi selengkapnya tentang nilai yang didukung, lihat kolom `Language tag` dalam tabel di topik Product Behavior di MSDN.

> **Note**
>
> Nilai `div`, `div-MV`, `hu`, dan `hu-HU` tidak didukung.

7. (Opsional) Untuk `TimeZoneKind`, masukkan `Local` atau `UTC`. Anda dapat mengatur ini untuk memberikan informasi zona waktu ketika tidak ada informasi zona waktu yang disertakan dalam stempel waktu log Anda. Jika parameter ini dibiarkan kosong dan jika stempel waktu Anda tidak menyertakan informasi zona waktu, CloudWatch Log default ke zona waktu lokal. Parameter ini diabaikan jika stempel waktu Anda sudah berisi informasi zona waktu.
8. (Opsional) Untuk `LineCount`, ketik jumlah baris di header untuk mengidentifikasi berkas log. Sebagai contoh, berkas log IIS memiliki header yang hampir identik. Anda bisa memasukkan **5**, yang akan membaca lima baris pertama header berkas log untuk mengidentifikasinya. Dalam berkas log IIS, baris ketiga adalah tanggal dan stempel waktu, tetapi stempel waktu tidak selalu dijamin akan berbeda antara berkas log. Untuk alasan ini, sebaiknya sertakan setidaknya satu baris data log aktual untuk membuat sidik jari berkas log secara unik.

Langkah 4: Konfigurasi kontrol aliran

Setiap tipe data harus memiliki tujuan yang sesuai di bagian `Flows`. Misalnya, untuk mengirim log kustom, log ETW, dan log sistem ke CloudWatch Log, tambahkan `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` ke `Flows` bagian.

> **Warning**
>
> Menambahkan langkah yang tidak valid akan memblokir aliran. Misalnya, jika Anda menambahkan langkah metrik disk, tetapi instans Anda tidak memiliki disk, semua langkah dalam aliran akan diblokir.

Anda dapat mengirim berkas log yang sama ke lebih dari satu tujuan. Misalnya, untuk mengirim log aplikasi ke dua tujuan yang berbeda yang Anda tetapkan di bagian `CloudWatchLogs`, tambahkan `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` ke bagian `Flows`.

Untuk mengonfigurasi kontrol aliran

1. Di file `AWS.EC2.Windows.CloudWatch.json`, temukan bagian `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. Untuk `Flows`, tambahkan setiap tipe data yang akan diunggah (misalnya, `ApplicationEventLog`) dan tujuannya (misalnya, `CloudWatchLogs`).

Langkah 5: Simpan konten JSON

Anda sekarang sudah selesai mengedit file JSON. Simpan file, dan tempel isi file ke editor teks di jendela lain. Anda akan membutuhkan isi file di langkah selanjutnya dalam prosedur ini.

Membuat peran IAM untuk Systems Manager

IAM role untuk kredensial instans diperlukan ketika Anda menggunakan Systems Manager Run Command. Peran ini memungkinkan Systems Manager untuk melakukan tindakan di instans. Untuk informasi selengkapnya, lihat Mengonfigurasi Peran Keamanan untuk Systems Manager di Panduan Pengguna AWS Systems Manager . Untuk informasi tentang cara melampirkan peran IAM ke instans yang ada, lihat Melampirkan Peran IAM ke Instans di Panduan Pengguna Amazon EC2.

Memverifikasi prasyarat Systems Manager

Sebelum Anda menggunakan Systems Manager Run Command untuk mengonfigurasi integrasi dengan CloudWatch Log, verifikasi bahwa instance Anda memenuhi persyaratan minimum. Untuk informasi selengkapnya, silakan lihat Prasyarat Systems Manager di Panduan Pengguna AWS Systems Manager .

Memverifikasi Akses Internet

Instans Amazon EC2 Windows Server dan instans terkelola harus memiliki akses internet keluar untuk mengirim data log dan peristiwa ke. CloudWatch Untuk informasi selengkapnya tentang cara mengonfigurasi akses internet, silakan lihat Gateway Internet dalam Panduan Pengguna VPC Amazon.

Aktifkan CloudWatch Log menggunakan Systems Manager Run Command

Run Command memungkinkan Anda mengelola konfigurasi instans Anda sesuai permintaan. Anda menentukan dokumen Manajer Sistem, menentukan parameter, dan mengeksekusi perintah pada satu atau beberapa instans. SSM agent di instans memproses perintah dan mengonfigurasi instans seperti yang ditentukan.

Untuk mengkonfigurasi integrasi dengan CloudWatch Log menggunakan Run Command

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Buka konsol SSM di https://console.aws.amazon.com/systems-manager/.
3. Di panel navigasi, pilih Jalankan Perintah.
4. Pilih Run a command (Jalankan perintah).
5. Untuk dokumen Command, pilih AWS- ConfigureCloudWatch.
6. Untuk instance Target, pilih instance yang akan diintegrasikan dengan CloudWatch Log. Jika Anda tidak melihat instans dalam daftar ini, instans tersebut mungkin tidak dikonfigurasi untuk Run Command. Untuk informasi selengkapnya, lihat Prasyarat Systems Manager di Panduan Pengguna Amazon EC2.
7. Untuk Status, pilih Enabled (Diaktifkan).
8. Untuk Properties (Properti), salin dan tempel konten JSON yang Anda buat dalam tugas sebelumnya.
9. Selesaikan bidang opsional yang lainnya dan pilih Run (Jalankan).

Gunakan prosedur berikut untuk melihat hasil eksekusi perintah di konsol Amazon EC2.

Untuk melihat output perintah di konsol

1. Pilih perintah.
2. Pilih tab Output.
3. Pilih View Output (Lihat Output). Halaman output perintah menampilkkan hasil eksekusi perintah Anda.

Mulai Cepat: Aktifkan instans Amazon EC2 Anda yang menjalankan Windows Server 2012 dan Windows Server 2008 untuk mengirim log ke Log CloudWatch

> **Tip**
>
> CloudWatch menyertakan agen terpadu baru yang dapat mengumpulkan log dan metrik dari instans EC2 dan server lokal. Kami menyarankan Anda menggunakan agen terpadu CloudWatch yang lebih baru. Untuk informasi selengkapnya, lihat Memulai dengan CloudWatch Log.
> Sisa bagian ini menjelaskan penggunaan agen CloudWatch Log yang lebih tua.

Aktifkan instans Amazon EC2 Anda yang menjalankan Windows Server 2012 dan Windows Server 2008 untuk mengirim log ke Log CloudWatch

Gunakan langkah-langkah berikut untuk mengaktifkan instance Anda yang menjalankan Windows Server 2012 dan Windows Server 2008 untuk mengirim CloudWatch log ke Log.

Unduh file konfigurasi contoh

Unduh file JSON contoh berikut ke komputer Anda: `AWS.EC2.Windows.CloudWatch.json`. Anda akan mengeditnya dalam langkah-langkah berikut.

Konfigurasikan file JSON untuk CloudWatch

Anda menentukan log mana yang akan dikirim CloudWatch dengan menentukan pilihan Anda di file konfigurasi JSON. Proses untuk membuat file ini dan menentukan pilihan Anda dapat memakan waktu 30 menit atau lebih untuk diselesaikan. Setelah Anda menyelesaikan tugas ini satu kali, Anda dapat menggunakan kembali file konfigurasi di semua instans Anda.

Langkah-langkah

- Langkah 1: Aktifkan CloudWatch Log
- Langkah 2: Konfigurasikan pengaturan untuk CloudWatch
- Langkah 3: Konfigurasi data untuk mengirim
- Langkah 4: Konfigurasi kontrol aliran

Langkah 1: Aktifkan CloudWatch Log

Di bagian atas file JSON, ubah "false" menjadi "true" untuk `IsEnabled`:

```
"IsEnabled": true,
```

Langkah 2: Konfigurasikan pengaturan untuk CloudWatch

Tentukan kredensial, Wilayah, nama grup log, dan namespace pengaliran log. Hal ini memungkinkan instance untuk mengirim data log ke CloudWatch Log. Untuk mengirim data log yang sama ke lokasi yang berbeda, Anda dapat menambahkan bagian tambahan dengan ID unik (misalnya, "CloudWatchLogs2" dan CloudWatchLogs 3") dan Wilayah yang berbeda untuk setiap ID.

Untuk mengkonfigurasi pengaturan untuk mengirim data log ke CloudWatch Log

1. Dalam file JSON, temukan bagian `CloudWatchLogs`.

   ```
   {
       "Id": "CloudWatchLogs",
       "FullName":
    "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
       "Parameters": {
           "AccessKey": "",
           "SecretKey": "",
           "Region": "us-east-1",
           "LogGroup": "Default-Log-Group",
           "LogStream": "{instance_id}"
       }
   },
   ```

2. Biarkan bidang `AccessKey` dan `SecretKey` tetap kosong. Anda mengonfigurasi kredensial menggunakan IAM role.
3. Untuk `Region`, ketik Wilayah untuk mengirim data log (misalnya, `us-east-2`).

4. Untuk `LogGroup`, ketik nama untuk grup log Anda. Nama ini muncul di layar Grup Log di CloudWatch konsol.
5. Untuk `LogStream`, ketik pengaliran log tujuan. Nama ini muncul di layar Grup Log > Streams di CloudWatch konsol.

   Jika Anda menggunakan `{instance_id}`, yaitu default-nya, nama pengaliran log adalah ID instans dari instans ini.

   Jika Anda menentukan nama aliran log yang belum ada, CloudWatch Log secara otomatis membuatnya untuk Anda. Anda dapat menentukan nama pengaliran log menggunakan string literal, variabel yang telah ditetapkan `{instance_id}`, `{hostname}`, dan `{ip_address}`, atau kombinasinya.

Langkah 3: Konfigurasi data untuk mengirim

Anda dapat mengirim data log peristiwa, data Event Tracing for Windows (ETW), dan data log lainnya ke CloudWatch Log.

Untuk mengirim data log peristiwa aplikasi Windows ke CloudWatch Log

1. Dalam file JSON, temukan bagian `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.
   - **4** - Unggah hanya pesan informasi.

Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim data log keamanan ke CloudWatch Log

1. Dalam file JSON, temukan bagian `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Untuk `Levels`, ketik **7** untuk mengunggah semua pesan.

Untuk mengirim data log peristiwa sistem ke CloudWatch Log

1. Dalam file JSON, temukan bagian `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.

- **4** - Unggah hanya pesan informasi.

Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim jenis data log peristiwa lainnya ke CloudWatch Log

1. Dalam file JSON, tambahkan bagian baru. Setiap bagian harus memiliki `Id` yang unik.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Untuk `Id`, ketik nama untuk log yang akan diunggah (misalnya, **`WindowsBackup`**).
3. Untuk `LogName`, ketik nama log yang akan diunggah. Anda dapat menemukan nama log sebagai berikut.

   a. Buka Event Viewer.

   b. Di panel navigasi, pilih Applications and Services Logs (Log Aplikasi dan Layanan).

   c. Buka log, lalu pilih Actions (Tindakan), Properties (Properti).
4. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.
   - **4** - Unggah hanya pesan informasi.

   Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim Event Tracing untuk data Windows ke CloudWatch Log

ETW (Event Tracing for Windows) menyediakan mekanisme pencatatan log yang efisien dan terperinci yang dapat digunakan aplikasi untuk menuliskan log. Setiap ETW dikendalikan oleh manajer sesi yang dapat memulai dan menghentikan sesi pencatatan. Setiap sesi memiliki penyedia dan satu atau beberapa konsumen.

1. Dalam file JSON, temukan bagian `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Untuk `LogName`, ketik nama log yang akan diunggah.
3. Untuk `Levels`, tentukan jenis pesan yang akan diunggah. Anda dapat menentukan salah satu nilai berikut:

   - **1** - Unggah hanya pesan kesalahan.
   - **2** - Unggah hanya pesan peringatan.
   - **4** - Unggah hanya pesan informasi.

   Anda dapat menggabungkan nilai-nilai untuk menyertakan lebih dari satu jenis pesan. Misalnya, nilai **3** mengunggah pesan kesalahan (**1**) dan pesan peringatan (**2**). Nilai **7** mengunggah pesan kesalahan (**1**), pesan peringatan (**2**), dan pesan informasi (**4**).

Untuk mengirim log khusus (file log berbasis teks apa pun) ke Log CloudWatch

1. Dalam file JSON, temukan bagian `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
```

```
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Untuk `LogDirectoryPath`, ketik jalur tempat log disimpan di instans Anda.
3. Untuk `TimestampFormat`, ketik format stempel waktu yang akan digunakan. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik Custom Date and Time Format Strings di MSDN.

> **Important**
>
> Berkas log sumber Anda harus memiliki stempel waktu di awal setiap baris log dan harus ada spasi setelah stempel waktu.

4. Untuk `Encoding`, ketik pengodean file yang akan digunakan (misalnya, UTF-8). Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik Encoding Class di MSDN.

> **Note**
>
> Gunakan nama pengodean, bukan nama tampilan.

5. (Opsional) Untuk `Filter`, ketik prefiks nama log. Biarkan parameter ini kosong untuk memantau semua file. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik FileSystemWatcherFilter Properti di MSDN.
6. (Opsional) Untuk `CultureName`, ketik lokal tempat stempel waktu dicatat. Jika `CultureName` kosong, default-nya adalah lokal yang sama yang saat ini digunakan oleh instans Windows Anda. Untuk informasi selengkapnya tentang nilai yang didukung, lihat kolom `Language tag` dalam tabel di topik Product Behavior di MSDN.

> **Note**
>
> Nilai `div`, `div-MV`, `hu`, dan `hu-HU` tidak didukung.

7. (Opsional) Untuk `TimeZoneKind`, ketik `Local` atau `UTC`. Anda dapat mengatur ini untuk memberikan informasi zona waktu ketika tidak ada informasi zona waktu yang disertakan dalam stempel waktu log Anda. Jika parameter ini dibiarkan kosong dan jika stempel waktu Anda tidak menyertakan informasi zona waktu, CloudWatch Log default ke zona waktu lokal. Parameter ini diabaikan jika stempel waktu Anda sudah berisi informasi zona waktu.
8. (Opsional) Untuk `LineCount`, ketik jumlah baris di header untuk mengidentifikasi berkas log. Sebagai contoh, berkas log IIS memiliki header yang hampir identik. Anda bisa memasukkan **5**, yang akan membaca tiga baris pertama header berkas log untuk mengidentifikasinya. Dalam berkas log IIS, baris ketiga adalah tanggal dan stempel waktu, tetapi stempel waktu tidak selalu dijamin akan berbeda antara berkas log. Untuk alasan ini, sebaiknya sertakan setidaknya satu baris data log aktual untuk membuat sidik jari berkas log secara unik.

Untuk mengirim data log IIS ke CloudWatch Log

1. Dalam file JSON, temukan bagian `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Untuk `LogDirectoryPath`, ketik folder tempat IIS log disimpan untuk situs individual (misalnya, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> Hanya format log W3C yang didukung. Format IIS, NCSA, dan Custom tidak didukung.

3. Untuk `TimestampFormat`, ketik format stempel waktu yang akan digunakan. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik Custom Date and Time Format Strings di MSDN.
4. Untuk `Encoding`, ketik pengodean file yang akan digunakan (misalnya, UTF-8). Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik Encoding Class di MSDN.

   > **Note**
   >
   > Gunakan nama pengodean, bukan nama tampilan.

5. (Opsional) Untuk `Filter`, ketik prefiks nama log. Biarkan parameter ini kosong untuk memantau semua file. Untuk informasi selengkapnya tentang nilai yang didukung, lihat topik FileSystemWatcherFilter Properti di MSDN.
6. (Opsional) Untuk `CultureName`, ketik lokal tempat stempel waktu dicatat. Jika `CultureName` kosong, default-nya adalah lokal yang sama yang saat ini digunakan oleh instans Windows Anda. Untuk informasi selengkapnya tentang nilai yang didukung, lihat kolom `Language tag` dalam tabel di topik Product Behavior di MSDN.

   > **Note**
   >
   > Nilai `div`, `div-MV`, `hu`, dan `hu-HU` tidak didukung.

7. (Opsional) Untuk `TimeZoneKind`, masukkan `Local` atau `UTC`. Anda dapat mengatur ini untuk memberikan informasi zona waktu ketika tidak ada informasi zona waktu yang disertakan dalam stempel waktu log Anda. Jika parameter ini dibiarkan kosong dan jika stempel waktu Anda tidak menyertakan informasi zona waktu, CloudWatch Log default ke zona waktu lokal. Parameter ini diabaikan jika stempel waktu Anda sudah berisi informasi zona waktu.
8. (Opsional) Untuk `LineCount`, ketik jumlah baris di header untuk mengidentifikasi berkas log. Sebagai contoh, berkas log IIS memiliki header yang hampir identik. Anda bisa memasukkan **5**, yang akan membaca lima baris pertama header berkas log untuk mengidentifikasinya. Dalam berkas log IIS, baris ketiga adalah tanggal dan stempel waktu, tetapi stempel waktu tidak selalu dijamin akan berbeda antara berkas log. Untuk alasan ini, sebaiknya sertakan setidaknya satu baris data log aktual untuk membuat sidik jari berkas log secara unik.

Langkah 4: Konfigurasi kontrol aliran

Setiap tipe data harus memiliki tujuan yang sesuai di bagian `Flows`. Misalnya, untuk mengirim log kustom, log ETW, dan log sistem ke CloudWatch Log, tambahkan `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` ke `Flows` bagian.

> **Warning**
>
> Menambahkan langkah yang tidak valid akan memblokir aliran. Misalnya, jika Anda menambahkan langkah metrik disk, tetapi instans Anda tidak memiliki disk, semua langkah dalam aliran akan diblokir.

Anda dapat mengirim berkas log yang sama ke lebih dari satu tujuan. Misalnya, untuk mengirim log aplikasi ke dua tujuan yang berbeda yang Anda tetapkan di bagian `CloudWatchLogs`, tambahkan `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` ke bagian `Flows`.

Untuk mengonfigurasi kontrol aliran

1. Di file `AWS.EC2.Windows.CloudWatch.json`, temukan bagian `Flows`.

   ```
   "Flows": {
       "Flows": [
         "PerformanceCounter,CloudWatch",
         "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
         "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
         "CustomLogs, CloudWatchLogs2",
         "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
       ]
   }
   ```

2. Untuk `Flows`, tambahkan setiap tipe data yang akan diunggah (misalnya, `ApplicationEventLog`) dan tujuannya (misalnya, `CloudWatchLogs`).

Anda sekarang sudah selesai mengedit file JSON. Anda akan menggunakannya di langkah berikutnya.

Memulai agen

Untuk mengaktifkan instans Amazon EC2 yang menjalankan Windows Server 2012 atau Windows Server 2008 untuk mengirim CloudWatch log ke Log, gunakan layanan EC2config (.

`EC2Config.exe`) Instans Anda harus memiliki EC2Config 4.0 atau yang lebih baru, dan Anda dapat menggunakan prosedur ini. Untuk informasi selengkapnya tentang menggunakan versi EC2config yang lebih lama, lihat Menggunakan EC2config 3.x atau Sebelumnya untuk Mengonfigurasi di Panduan Pengguna Amazon EC2 CloudWatch

Untuk mengkonfigurasi CloudWatch menggunakan EC2config 4.x

1. Periksa pengodean file `AWS.EC2.Windows.CloudWatch.json` yang Anda edit sebelumnya dalam prosedur ini. Hanya pengodean UTF-8 tanpa BOM yang didukung. Kemudian simpan file di folder berikut di instans Windows Server 2008 - 2012 R2: `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.
2. Mulai atau mulai ulang agen SSM (`AmazonSSMAgent.exe`) menggunakan panel kontrol Layanan Windows atau menggunakan PowerShell perintah berikut:

```
PS C:\> Restart-Service AmazonSSMAgent
```

Setelah agen SSM restart, ia mendeteksi file konfigurasi dan mengkonfigurasi instance untuk integrasi. CloudWatch Jika Anda mengubah parameter dan pengaturan dalam file konfigurasi lokal, Anda perlu memulai ulang SSM agent untuk mengikuti perubahannya. Untuk menonaktifkan CloudWatch integrasi pada instance, ubah `IsEnabled` ke `false` dan simpan perubahan Anda dalam file konfigurasi.

Mulai Cepat: Instal agen CloudWatch Log menggunakan AWS OpsWorks dan Chef

Anda dapat menginstal agen CloudWatch Log dan membuat aliran log menggunakan AWS OpsWorks dan Chef, yang merupakan sistem pihak ketiga dan alat otomatisasi infrastruktur cloud. Chef menggunakan "resep", yang Anda tulis untuk menginstal dan mengonfigurasi perangkat lunak di komputer Anda, dan "buku resep," yang merupakan kumpulan resep, untuk melakukan konfigurasi dan tugas distribusi kebijakannya. Untuk informasi selengkapnya, lihat Chef.

Contoh resep Chef di bawah ini menunjukkan cara memantau satu berkas log di setiap instans EC2. Resep menggunakan nama tumpukan sebagai grup log dan nama host instans sebagai nama pengaliran log. Untuk memantau beberapa berkas log, Anda perlu memperluas resep untuk membuat beberapa grup log dan pengaliran log.

Langkah 1: Buat resep khusus

Buat repositori untuk menyimpan resep Anda. AWS OpsWorks mendukung Git dan Subversion, atau Anda dapat menyimpan arsip di Amazon S3. Struktur repositori buku resep Anda dijelaskan dalam Repositori Cookbook di Panduan Pengguna AWS OpsWorks . Contoh di bawah ini mengasumsikan bahwa buku resep bernama `logs`. Resep install.rb menginstal agen Log. CloudWatch Anda juga dapat mengunduh contoh buku masak (CloudWatchLogs-Cookbooks.zip).

Buat file bernama metadata.rb yang berisi kode berikut:

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

Buat file konfigurasi CloudWatch Log:

```
#config.rb

template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Unduh dan instal agen CloudWatch Log:

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end

remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py"
  mode "0755"
end
```

```
  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> Dalam contoh di atas, ganti *region* dengan salah satu dari yang berikut: us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, atau sa-east-1.

Jika instalasi agen gagal, periksa untuk memastikan bahwa paket `python-dev` sudah diinstal. Jika belum, gunakan perintah berikut, lalu coba lagi instalasi agen:

```
sudo apt-get -y install python-dev
```

Resep ini menggunakan file templat cwlogs.cfg.erb yang dapat Anda modifikasi untuk menentukan berbagai atribut seperti file apa yang akan dicatat. Untuk informasi selengkapnya tentang atribut ini, lihat CloudWatch Referensi agen log.

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state


## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
```

```
#
#log_stream_name = {instance_id}
#
## Format specifier for timestamp parsing.
#
#datetime_format = %b %d %H:%M:%S
#
#

[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

Templat mendapat nama tumpukan dan nama host dengan referensi atribut yang sesuai dalam konfigurasi tumpukan dan deployment JSON. Atribut yang menentukan file yang akan dicatat ditentukan dalam file atribut cwlogs cookbook default.rb (logs/attributes/default.rb).

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

Langkah 2: Buat AWS OpsWorks tumpukan

1. Buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di OpsWorks Dasbor, pilih Tambahkan tumpukan untuk membuat AWS OpsWorks tumpukan.
3. Di layar Add stack (Tambah tumpukan), pilih Chef 11 stack (Tumpukan Chef 11).
4. Untuk Stack name (Nama tumpukan), masukkan nama.
5. Untuk Use custom Chef Cookbooks (Gunakan Chef Cookboks khusus), pilih Yes (Ya).
6. Untuk Repository type (Jenis repositori), pilih jenis repositori yang Anda gunakan. Jika Anda menggunakan contoh di atas, pilih Http Archive (Arsip Http).
7. Untuk Repository URL (URL Repositori), masukkan repositori tempat Anda menyimpan buku resep yang Anda buat di langkah sebelumnya. Jika Anda menggunakan contoh di atas, masukkan **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`**.
8. Pilih Add Stack (Tambah tumpukan) untuk membuat tumpukan.

Langkah 3: Perluas IAM role Anda

Untuk menggunakan CloudWatch Log dengan AWS OpsWorks instans Anda, Anda perlu memperluas peran IAM yang digunakan oleh instance Anda.

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Policies (Kebijakan), Create Policy (Buat Kebijakan).
3. Di halaman Create Policy (Buat Kebijakan), di bawah Create Your Own Policy (Buat Kebijakan Anda Sendiri), pilih Select (Pilihan). Untuk informasi selengkapnya tentang membuat kebijakan khusus, lihat Kebijakan IAM untuk Amazon EC2 di Panduan Pengguna Amazon EC2.
4. Di halaman Review Policy (Tinjau Kebijakan), untuk Policy Name (Nama Kebijakan), ketikkan nama untuk kebijakan tersebut.
5. Untuk Policy Document (Dokumen Kebijakan), tempelkan kebijakan berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

6. Pilih Buat Kebijakan.
7. Di panel navigasi, pilih Peran, lalu di panel konten, untuk Nama Peran, pilih nama peran instance yang digunakan oleh tumpukan Anda AWS OpsWorks . Anda dapat menemukan peran yang digunakan oleh tumpukan Anda di pengaturan tumpukan (default-nya adalah `aws-opsworks-ec2-role`).

> **Note**
>
> Pilih nama peran, bukan kotak centang.

8. Di tab Permissions (Izin), di bawah Managed Policies (Kebijakan Terkelola), pilih Attach Policy (Lampirkan Kebijakan.).
9. Di halaman Attach Policy (Lampirkan Kebijakan), di header tabel (di sebelah Filter dan Search (Pencarian)), pilih Policy Type (Tipe Kebijakan), Customer Managed Policies (Kebijakan yang Dikelola Pelanggan).
10. Untuk Customer Managed Policies (Kebijakan yang Dikelola Pelanggan), pilih kebijakan IAM yang Anda buat di atas dan pilih Attach Policy (Lampirkan Kebijakan).

    Untuk informasi selengkapnya tentang pengguna dan kebijakan, lihat Pengguna dan Grup IAM dan Mengelola Kebijakan IAM di Panduan Pengguna IAM.

Langkah 4: Tambahkan lapisan

1. Buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di panel navigasi, pilih Layers (Lapisan).
3. Di panel konten, pilih lapisan dan pilih Add layer (Tambah lapisan).
4. Pada OpsWorkstab, untuk tipe Layer, pilih Custom.
5. Untuk Name (Nama) dan Short name (Nama pendek), masukkan nama panjang dan pendek untuk lapisan, lalu pilih Add layer (Tambah lapisan).
6. Pada tab Resep, di bawah Resep Koki Kustom, ada beberapa judul— Setup, Configure, Deploy, Undeploy, dan Shutdown —yang sesuai dengan peristiwa siklus hidup. AWS OpsWorks AWS OpsWorks memicu peristiwa ini pada titik-titik penting ini dalam siklus hidup instance, yang menjalankan resep terkait.

   > **Note**
   >
   > Jika judul di atas tidak terlihat, di bawah Custom Chef Recipes (Resep Chef Khusus), pilih edit.

7. Masukkan logs::config, logs::install di sebelah Setup (Penyiapan), pilih + untuk menambahkannya ke daftar, lalu pilih Save (Simpan).

AWS OpsWorks menjalankan resep ini pada setiap instance baru di layer ini, tepat setelah instance boot.

Langkah 5: Tambahkan instans

Lapisan hanya mengontrol cara mengonfigurasi instans. Anda sekarang perlu menambahkan beberapa instans ke lapisan dan memulainya.

1. Buka AWS OpsWorks konsol di https://console.aws.amazon.com/opsworks/.
2. Di panel navigasi, pilih Instances (Instans), lalu di lapisan Anda, pilih + Instance (+ Instans).
3. Setujui pengaturan default dan pilih Add Instance (Tambah Instans) untuk menambahkan instans ke lapisan.
4. Di kolom baris Actions (Tindakan), klik start (mulai) untuk memulai instans.

   AWS OpsWorks meluncurkan instans EC2 baru dan mengonfigurasi CloudWatch Log. Status instans berubah menjadi online (daring) ketika sudah siap.

Langkah 6: Lihat log Anda

Anda akan melihat grup log dan aliran log yang baru dibuat di CloudWatch konsol setelah agen berjalan selama beberapa saat.

Untuk informasi selengkapnya, lihat Lihat data log yang dikirim ke CloudWatch Log.

Laporkan status agen CloudWatch Log

Gunakan prosedur berikut untuk melaporkan status agen CloudWatch Log pada instans EC2 Anda.

Untuk melaporkan status agen

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat Connect to Your Instance di Panduan Pengguna Amazon EC2.

   Untuk informasi selengkapnya tentang masalah koneksi, lihat Memecahkan Masalah Menyambung ke Instans Anda di Panduan Pengguna Amazon EC2
2. Di jendela perintah, jalankan perintah berikut:

```
sudo service awslogs status
```

Jika Anda menjalankan Amazon Linux 2, ketik perintah berikut:

```
sudo service awslogsd status
```

3. Periksa file /var/log/awslogs.log untuk setiap kesalahan, peringatan, atau masalah dengan agen CloudWatch Log.

Mulai agen CloudWatch Log

Jika agen CloudWatch Log pada instans EC2 Anda tidak memulai secara otomatis setelah instalasi, atau jika Anda menghentikan agen, Anda dapat menggunakan prosedur berikut untuk memulai agen.

Untuk memulai agen

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat Connect to Your Instance di Panduan Pengguna Amazon EC2.

   Untuk informasi selengkapnya tentang masalah koneksi, lihat Pemecahan Masalah Menghubungkan ke Instans Anda di Panduan Pengguna Amazon EC2.
2. Di jendela perintah, jalankan perintah berikut:

   ```
   sudo service awslogs start
   ```

   Jika Anda menjalankan Amazon Linux 2, ketik perintah berikut:

   ```
   sudo service awslogsd start
   ```

Hentikan agen CloudWatch Log

Gunakan prosedur berikut untuk menghentikan agen CloudWatch Log pada instans EC2 Anda.

Untuk menghentikan agen

1. Connect ke instans EC2 Anda. Untuk informasi selengkapnya, lihat Connect to Your Instance di Panduan Pengguna Amazon EC2.

Untuk informasi selengkapnya tentang masalah koneksi, lihat Pemecahan Masalah Menghubungkan ke Instans Anda di Panduan Pengguna Amazon EC2.

2. Di jendela perintah, jalankan perintah berikut:

```
sudo service awslogs stop
```

Jika Anda menjalankan Amazon Linux 2, ketik perintah berikut:

```
sudo service awslogsd stop
```

Mulai Cepat: Gunakan AWS CloudFormation untuk memulai dengan CloudWatch Log

AWS CloudFormation memungkinkan Anda untuk mendeskripsikan dan menyediakan AWS sumber daya Anda dalam format JSON. Keuntungan dari metode ini termasuk mampu mengelola kumpulan AWS sumber daya sebagai satu unit, dan dengan mudah mereplikasi AWS sumber daya Anda di seluruh Wilayah.

Saat Anda menyediakan AWS penggunaan AWS CloudFormation, Anda membuat templat yang menjelaskan AWS sumber daya yang akan digunakan. Contoh berikut adalah cuplikan templat yang membuat grup log dan filter metrik yang menghitung 404 kemunculan dan mengirimkan jumlah ini ke grup log.

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code =
 404, size, ...]",
```

```
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    }
}
```

Ini adalah contoh dasar. Anda dapat mengatur penerapan CloudWatch Log yang jauh lebih kaya menggunakan. AWS CloudFormation Untuk informasi selengkapnya tentang contoh templat, lihat Cuplikan Templat CloudWatch Log Amazon di AWS CloudFormation Panduan Pengguna. Untuk informasi selengkapnya tentang memulai, lihat Memulai AWS CloudFormation dalam Panduan Pengguna AWS CloudFormation .

Menggunakan CloudWatch Log dengan AWS SDK

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

Dokumentasi SDK	Contoh kode
AWS SDK for C++	AWS SDK for C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK for Go	AWS SDK for Go contoh kode
AWS SDK for Java	AWS SDK for Java contoh kode
AWS SDK for JavaScript	AWS SDK for JavaScript contoh kode
AWS SDK for Kotlin	AWS SDK for Kotlin contoh kode
AWS SDK for .NET	AWS SDK for .NET contoh kode
AWS SDK for PHP	AWS SDK for PHP contoh kode
AWS Tools for PowerShell	Alat untuk contoh PowerShell kode
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) contoh kode
AWS SDK for Ruby	AWS SDK for Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh khusus untuk CloudWatch Log, lihatContoh kode untuk CloudWatch Log menggunakan AWS SDK.

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Menganalisis data log dengan Wawasan CloudWatch Log

Dengan Wawasan CloudWatch Log, Anda dapat mencari dan menganalisis data log Anda secara interaktif di Log Amazon CloudWatch . Anda dapat melakukan kueri untuk membantu Anda agar lebih efisien dan efektif dalam menanggapi masalah operasional. Jika terjadi masalah, Anda dapat menggunakan Wawasan CloudWatch Log untuk mengidentifikasi penyebab potensial dan memvalidasi perbaikan yang diterapkan.

CloudWatch Logs Insights mencakup bahasa kueri yang dibuat khusus dengan beberapa perintah sederhana namun kuat. CloudWatch Logs Insights menyediakan contoh kueri, deskripsi perintah, pelengkapan otomatis kueri, dan penemuan bidang log untuk membantu Anda memulai. Kueri contoh disertakan untuk beberapa jenis log layanan AWS .

CloudWatch Logs Insights secara otomatis menemukan bidang dalam log dari AWS layanan seperti Amazon Route 53,, AWS Lambda AWS CloudTrail, dan Amazon VPC, dan aplikasi atau log kustom apa pun yang memancarkan peristiwa log sebagai JSON.

Anda dapat menggunakan Wawasan CloudWatch Log untuk mencari data log yang dikirim ke CloudWatch Log pada 5 November 2018 atau lebih baru.

> **Important**
>
> CloudWatch Wawasan Log tidak dapat mengakses peristiwa log dengan stempel waktu yang mendahului waktu pembuatan grup log.

Anda juga dapat menggunakan bahasa alami untuk membuat kueri Wawasan CloudWatch Log. Untuk melakukan hal itu, ajukan pertanyaan atau jelaskan data yang Anda cari. Kemampuan berbantuan AI ini menghasilkan kueri berdasarkan prompt Anda dan memberikan line-by-line penjelasan tentang cara kerja kueri. Untuk informasi selengkapnya, lihat Menggunakan bahasa alami untuk membuat dan memperbarui kueri Wawasan CloudWatch Log.

Jika Anda masuk ke akun yang disiapkan sebagai akun pemantauan dalam pengamatan CloudWatch lintas akun, Anda dapat menjalankan kueri Wawasan CloudWatch Log pada grup log di akun sumber yang ditautkan ke akun pemantauan ini. Anda dapat menjalankan kueri yang menanyakan beberapa grup log yang terletak di akun yang berbeda. Untuk informasi lebih lanjut, lihat CloudWatch observabilitas lintas akun.

Satu permintaan dapat menanyakan hingga 50 grup log. Waktu kueri habis setelah 60 menit, jika belum selesai. Hasil kueri tersedia selama 7 hari.

Anda dapat menyimpan kueri yang telah Anda buat. Hal ini dapat membantu Anda menjalankan kueri yang kompleks ketika diperlukan tanpa harus membuat ulang setiap kali Anda ingin menjalankannya.

CloudWatch Kueri Log Insights dikenakan biaya berdasarkan jumlah data yang ditanyakan. Untuk informasi selengkapnya, lihat CloudWatch Harga Amazon.

> **Important**
>
> Jika tim keamanan jaringan Anda tidak mengizinkan penggunaan soket web, saat ini Anda tidak dapat mengakses bagian CloudWatch Logs Insights dari CloudWatch konsol. Anda dapat menggunakan kemampuan kueri CloudWatch Log Insights menggunakan API. Untuk informasi selengkapnya, lihat StartQuerydi Referensi API Amazon CloudWatch Logs.

Daftar Isi

Perintah yang didukung di kelas log

Semua perintah kueri Wawasan CloudWatch Log didukung pada grup log di kelas log Standar. Grup log di kelas log Akses Jarang mendukung semua perintah kueri kecuali`pattern`,`diff`, dan`unmask`.

Memulai: Tutorial kueri

Bagian berikut mencakup contoh tutorial kueri untuk membantu Anda memulai dengan Wawasan CloudWatch Log.

Topik

- Tutorial: Jalankan dan modifikasi kueri sampel
- Tutorial: Jalankan kueri dengan fungsi agregasi
- Tutorial: Jalankan kueri yang menghasilkan visualisasi yang dikelompokkan berdasarkan bidang log
- Tutorial: Jalankan kueri yang menghasilkan visualisasi deret waktu

Tutorial: Jalankan dan modifikasi kueri sampel

Tutorial berikut membantu Anda memulai dengan Wawasan CloudWatch Log. Anda menjalankan kueri sampel, lalu melihat cara memodifikasi dan menjalankannya kembali.

Untuk menjalankan kueri, Anda harus sudah memiliki log yang disimpan di CloudWatch Log. Jika Anda sudah menggunakan CloudWatch Log dan memiliki grup log dan aliran log yang disiapkan, Anda siap untuk memulai. Anda mungkin juga sudah memiliki log jika Anda menggunakan layanan seperti AWS CloudTrail, Amazon Route 53, atau Amazon VPC dan Anda telah menyiapkan log dari layanan tersebut untuk masuk ke CloudWatch Log. Untuk informasi selengkapnya tentang mengirim CloudWatch log ke Log, lihatMemulai dengan CloudWatch Log.

Kueri dalam Wawasan CloudWatch Log mengembalikan sekumpulan bidang dari peristiwa log atau hasil agregasi matematis atau operasi lain yang dilakukan pada peristiwa log. Tutorial ini menunjukkan kueri yang mengembalikan daftar log acara.

Jalankan kueri sampel

Untuk menjalankan kueri sampel Wawasan CloudWatch Log

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

2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.

   Pada halaman Wawasan Log, editor kueri berisi kueri default yang menampilkan 20 peristiwa log terbaru.

3. Dalam menu tarik-turun Pilih grup log, pilih satu atau beberapa grup log untuk kueri.

   Jika ini adalah akun pemantauan dalam pengamatan CloudWatch lintas akun, Anda dapat memilih grup log di akun sumber serta akun pemantauan. Satu kueri dapat menanyakan log dari akun yang berbeda sekaligus.

   Anda dapat memfilter grup log berdasarkan nama grup log, ID akun, atau label akun.

   Saat Anda memilih grup log di kelas log Standar, Wawasan CloudWatch Log secara otomatis mendeteksi bidang data dalam grup. Untuk melihat bidang yang ditemukan, pilih menu Fields di dekat kanan atas halaman.

   > **Note**
   >
   > Bidang yang ditemukan hanya didukung untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

4. (Opsional) Gunakan pemilih interval waktu untuk memilih periode waktu yang ingin Anda kueri.

   Anda dapat memilih antara interval 5 dan 30 menit; interval 1, 3, dan 12 jam; atau kerangka waktu khusus.

5. Pilih Jalankan untuk melihat hasilnya.

   Untuk tutorial ini, hasilnya mencakup 20 peristiwa log yang paling baru ditambahkan.

   CloudWatch Log menampilkan grafik batang peristiwa log dalam grup log dari waktu ke waktu. Grafik batang tidak hanya menunjukkan peristiwa dalam tabel, tetapi juga distribusi peristiwa dalam grup log yang cocok dengan kueri dan jangka waktu.

6. Untuk melihat semua bidang untuk peristiwa log yang dikembalikan, pilih ikon tarik-turun segitiga di sebelah kiri acara bernomor.

Ubah kueri sampel

Dalam tutorial ini, Anda mengubah kueri sampel untuk menunjukkan 50 log acara terbaru.

Jika Anda belum menjalankan tutorial sebelumnya, lakukan sekarang. Tutorial ini dimulai di tempat tutorial sebelumnya berakhir.

> **Note**
>
> Beberapa contoh kueri yang disediakan dengan penggunaan CloudWatch Log Insights `head` atau `tail` perintah sebagai gantinya. `limit` Perintah ini akan tidak digunakan lagi dan telah diganti dengan `limit`. Gunakan `limit`, dan bukan `head` atau `tail` dalam semua kueri yang Anda tulis.

Untuk memodifikasi CloudWatch kueri sampel Wawasan Log

1. Di editor kueri, ubah 20 menjadi 50, lalu pilih Run (Jalankan).

   Hasil kueri baru akan muncul. Dengan asumsi ada cukup data dalam grup log selama rentang waktu default, sekarang ada 50 log acara yang tercantum.
2. (Opsional) Anda dapat menyimpan kueri yang telah Anda buat. Untuk menyimpan kueri ini, pilih Save (Simpan). Untuk informasi selengkapnya, lihat Simpan dan jalankan kembali kueri CloudWatch Logs Insights.

Tambahkan perintah filter ke kueri sampel

Tutorial ini menunjukkan cara membuat perubahan yang lebih kuat pada kueri di editor kueri. Dalam tutorial ini, Anda memfilter hasil kueri sebelumnya berdasarkan bidang dalam log acara yang diambil.

Jika Anda belum menjalankan tutorial sebelumnya, lakukan sekarang. Tutorial ini dimulai di tempat tutorial sebelumnya berakhir.

Untuk menambahkan perintah filter ke kueri sebelumnya

1. Tentukan bidang untuk memfilter. Untuk melihat bidang paling umum yang CloudWatch terdeteksi Log dalam peristiwa log yang terdapat dalam grup log yang dipilih dalam 15 menit terakhir, dan persentase peristiwa log di mana setiap bidang muncul, pilih Bidang di sisi kanan halaman.

   Untuk melihat bidang yang terdapat dalam log acara tertentu, pilih ikon di sebelah kiri baris tersebut.

Bidang awsRegion mungkin muncul dalam log acara Anda, tergantung pada kejadian yang ada di log Anda. Untuk bagian selanjutnya dalam tutorial ini, kita menggunakan awsRegion sebagai bidang filter, tetapi Anda dapat menggunakan bidang yang berbeda jika bidang tersebut tidak tersedia.

2. Di kotak editor kueri, tempatkan kursor Anda setelah 50, lalu tekan Enter.
3. Di baris baru, pertama masukkan | (karakter pipa) dan spasi. Perintah dalam kueri Wawasan CloudWatch Log harus dipisahkan oleh karakter pipa.
4. Masukkan **`filter awsRegion="us-east-1"`**.
5. Pilih Jalankan.

   Kueri berjalan lagi, dan sekarang menampilkan 50 hasil terbaru yang cocok dengan filter baru.

   Jika Anda memfilter dengan bidang yang berbeda dan mendapat hasil kesalahan, Anda mungkin perlu melakukan escape pada nama bidang. Jika nama bidang mengandung karakter non-alfanumerik, Anda harus menempatkan karakter backtick (`) sebelum dan sesudah nama bidang (misalnya, **`` `error-code`="102" ``**).

   Anda harus menggunakan karakter backtick untuk nama bidang yang berisi karakter non-alfanumerik, tetapi tidak untuk nilai. Nilai selalu ada dalam tanda kutip (").

CloudWatch Log Insights mencakup kemampuan kueri yang kuat, termasuk beberapa perintah dan dukungan untuk ekspresi reguler, matematika, dan operasi statistik. Untuk informasi selengkapnya, lihat CloudWatch Sintaks kueri Log Insights.

Tutorial: Jalankan kueri dengan fungsi agregasi

Anda dapat menggunakan fungsi agregasi dengan `stats` perintah dan sebagai argumen untuk fungsi lainnya. Dalam tutorial ini, Anda menjalankan perintah query yang menghitung jumlah peristiwa log yang berisi bidang tertentu. Perintah query mengembalikan jumlah total yang dikelompokkan berdasarkan nilai bidang tertentu atau nilai. Untuk informasi selengkapnya tentang fungsi agregasi, lihat Operasi dan fungsi yang didukung di Panduan Pengguna CloudWatch Log Amazon.

Untuk menjalankan kueri dengan fungsi agregasi

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Dalam menu tarik-turun Pilih grup log, pilih satu atau beberapa grup log untuk kueri.

Jika ini adalah akun pemantauan dalam pengamatan CloudWatch lintas akun, Anda dapat memilih grup log di akun sumber serta akun pemantauan. Satu kueri dapat menanyakan log dari akun yang berbeda sekaligus.

Anda dapat memfilter grup log berdasarkan nama grup log, ID akun, atau label akun.

Saat Anda memilih grup CloudWatch log, Wawasan Log secara otomatis mendeteksi bidang data dalam grup log jika itu adalah grup log kelas Standar. Untuk melihat bidang yang ditemukan, pilih menu Fields di dekat kanan atas halaman.

4. Hapus kueri default di editor kueri, dan masukkan perintah berikut:

```
stats count(*) by fieldName
```

5. Ganti ***fieldName*** dengan field yang ditemukan dari menu Fields.

   Menu Fields terletak di kanan atas halaman dan menampilkan semua bidang yang ditemukan yang mendeteksi Wawasan CloudWatch Log di grup log Anda.

6. Pilih Jalankan untuk melihat hasil kueri.

   Hasil kueri menunjukkan jumlah catatan dalam grup log Anda yang cocok dengan perintah kueri dan jumlah total yang dikelompokkan berdasarkan nilai atau nilai bidang yang ditentukan.

Tutorial: Jalankan kueri yang menghasilkan visualisasi yang dikelompokkan berdasarkan bidang log

Ketika menjalankan kueri yang menggunakan fungsi `stats` untuk mengelompokkan hasil yang dikembalikan oleh nilai dari satu atau beberapa bidang dalam entri log, Anda dapat melihat hasilnya sebagai diagram batang, diagram lingkaran, grafik garis, atau grafik area bertumpuk. Hal ini membantu memvisualisasikan tren dalam log Anda dengan lebih efisien.

Untuk menjalankan kueri untuk visualisasi

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Dalam menu tarik-turun Pilih grup log, pilih satu atau beberapa grup log untuk kueri.

Jika ini adalah akun pemantauan dalam pengamatan CloudWatch lintas akun, Anda dapat memilih grup log di akun sumber serta akun pemantauan. Satu kueri dapat menanyakan log dari akun yang berbeda sekaligus.

Anda dapat memfilter grup log berdasarkan nama grup log, ID akun, atau label akun.

4. Di editor kueri, hapus konten saat ini, masukkan fungsi `stats` berikut ini, lalu pilih Run query (Jalankan kueri).

```
stats count(*) by @logStream
    | limit 100
```

Hasilnya menunjukkan jumlah log acara dalam grup log untuk setiap pengaliran log. Hasilnya terbatas hanya 100 baris.

5. Pilih tab Visualization (Visualisasi).
6. Pilih panah di sebelah Line (Garis), lalu pilih Bar (Batang).

Akan muncul diagram batang yang menampilkan balok untuk setiap pengaliran log.

Tutorial: Jalankan kueri yang menghasilkan visualisasi deret waktu

Ketika menjalankan kueri yang menggunakan fungsi `bin()` untuk mengelompokkan hasil yang dikembalikan menurut jangka waktu, Anda dapat melihat hasilnya sebagai grafik garis, grafik area bertumpuk, diagram lingkaran, atau diagram batang. Hal ini membantu memvisualisasikan tren dalam log acara dengan lebih efisien dari waktu ke waktu.

Untuk menjalankan kueri untuk visualisasi

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Dalam menu tarik-turun Pilih grup log, pilih satu atau beberapa grup log untuk kueri.

Jika ini adalah akun pemantauan dalam pengamatan CloudWatch lintas akun, Anda dapat memilih grup log di akun sumber serta akun pemantauan. Satu kueri dapat menanyakan log dari akun yang berbeda sekaligus.

Anda dapat memfilter grup log berdasarkan nama grup log, ID akun, atau label akun.

4. Di editor kueri, hapus konten saat ini, masukkan fungsi `stats` berikut ini, lalu pilih Run query (Jalankan kueri).

   ```
   stats count(*) by bin(30s)
   ```

   Hasilnya menunjukkan jumlah peristiwa log dalam grup log yang diterima oleh CloudWatch Log untuk setiap periode 30 detik.

5. Pilih tab Visualization (Visualisasi).

   Hasilnya ditampilkan sebagai grafik garis. Untuk beralih ke diagram batang, diagram lingkaran, atau diagram area bertumpuk, pilih panah di samping Line (Garis) di kiri atas grafik.

Log yang didukung dan bidang yang ditemukan

CloudWatch Log Insights mendukung berbagai jenis log. Untuk setiap log yang dikirim ke grup log kelas Standar Amazon CloudWatch Logs, CloudWatch Logs Insights secara otomatis menghasilkan lima bidang sistem:

- `@message` berisi log acara mentah yang belum diurai. Ini setara dengan `message` bidang di InputLogevent.
- `@timestamp`berisi stempel waktu acara di bidang peristiwa log. `timestamp` Ini setara dengan `timestamp` bidang di InputLogevent.
- `@ingestionTime`berisi waktu ketika CloudWatch Log menerima peristiwa log.
- `@logStream` berisi nama pengaliran log yang ditambahi log acara. Log mengalirkan log grup melalui proses yang sama yang menghasilkannya.
- `@log` adalah pengidentifikasi grup log dalam bentuk *`account-id`*`:`*`log-group-name`*. Saat menanyakan beberapa grup log, ini dapat berguna untuk mengidentifikasi grup log mana yang termasuk dalam acara tertentu.

> **Note**
>
> Penemuan bidang hanya didukung untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

CloudWatch Logs Insights menyisipkan simbol @ di awal bidang yang dihasilkannya.

Untuk banyak jenis CloudWatch log, Log juga secara otomatis menemukan bidang log yang terdapat dalam log. Bidang penemuan otomatis ini ditunjukkan dalam tabel berikut.

Untuk jenis log lain dengan bidang yang tidak ditemukan secara otomatis oleh Wawasan CloudWatch Log, Anda dapat menggunakan `parse` perintah untuk mengekstrak dan membuat bidang yang diekstrak untuk digunakan dalam kueri tersebut. Untuk informasi selengkapnya, lihat CloudWatch Sintaks kueri Log Insights.

Jika nama bidang log yang ditemukan dimulai dengan @ karakter, Wawasan CloudWatch Log akan menampilkannya dengan tambahan yang @ ditambahkan ke awal. Sebagai contoh, jika nama bidang log adalah `@example.com`, nama bidang ini ditampilkan sebagai `@@example.com`.

Jenis log	Bidang log yang ditemukan
Log alur Amazon VPC	`@timestamp` , `@logStream` , `@message`, `accountId` , `endTime`, `interfaceId` , `logStatus` , `startTime` , `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`
Log Route 53	`@timestamp` , `@logStream` , `@message`, `edgeLocation` , `ednsClientSubnet` , `hostZoneId` , `protocol`, `queryName` , `queryTimestamp` , `queryType` , `resolverIp` , `responseCode` , `version`
Log Lambda	`@timestamp` , `@logStream` , `@message`, `@requestId` , `@duration`, `@billedDuration` , `@type`, `@maxMemoryUsed` , `@memorySize` Jika baris log Lambda berisi ID jejak X-Ray, itu juga mencakup bidang berikut: `@xrayTraceId` dan `@xraySegmentId` . CloudWatch Logs Insights secara otomatis menemukan bidang log di log Lambda, tetapi hanya untuk fragmen JSON pertama yang disematkan di setiap peristiwa log. Jika log acara Lambda berisi beberapa fragmen JSON, Anda dapat mengurai dan mengekstraksi bidang log menggunakan perintah **`parse`**. Untuk informasi selengkapnya, lihat Bidang di log JSON.
CloudTrail log Log dalam format JSON	Untuk informasi selengkapnya, lihat Bidang di log JSON.

Jenis log	Bidang log yang ditemukan
Jenis log lainnya	`@timestamp` , `@ingestionTime` , `@logStream` , `@message`, `@log`.

Bidang di log JSON

Dengan Wawasan CloudWatch Log, Anda menggunakan notasi titik untuk mewakili bidang JSON. Bagian ini berisi contoh peristiwa JSON dan cuplikan kode yang menunjukkan bagaimana Anda dapat mengakses bidang JSON menggunakan notasi titik.

Contoh: acara JSON

```
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
```

```
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
}
```

Contoh acara JSON berisi objek yang bernama`userIdentity`. `userIdentity`berisi bidang yang diberi nama`type`. Untuk mewakili nilai `type` menggunakan notasi titik, Anda menggunakan`userIdentity.type`.

Contoh acara JSON berisi array yang diratakan ke daftar nama bidang bersarang dan nilai. Untuk mewakili nilai `instanceId` untuk item pertama di`requestParameters.instancesSet`, Anda menggunakan`requestParameters.instancesSet.items.0.instanceId`. Angka `0` yang ditempatkan sebelum bidang `instanceID` mengacu pada posisi nilai untuk bidang`items`. Contoh berikut berisi cuplikan kode yang menunjukkan bagaimana Anda dapat mengakses bidang JSON bersarang dalam peristiwa log JSON.

Contoh: Query

```
fields @timestamp, @message
| filter requestParameters.instancesSet.items.0.instanceId="i-abcde123"
| sort @timestamp desc
```

Cuplikan kode menunjukkan kueri yang menggunakan notasi titik dengan `filter` perintah untuk mengakses nilai bidang JSON bersarang. `instanceId` Kueri menyaring pesan di mana nilai `instanceId` sama `"i-abcde123"` dan mengembalikan semua peristiwa log yang berisi nilai yang ditentukan.

> **Note**
>
> CloudWatch Wawasan Log dapat mengekstrak maksimal 200 bidang peristiwa log dari log JSON. Untuk bidang tambahan yang tidak diekstrak, Anda dapat menggunakan `parse`

perintah untuk mengekstrak bidang dari peristiwa log mentah yang tidak diurai di bidang pesan. Untuk informasi selengkapnya tentang `parse` perintah, lihat Sintaks kueri di Panduan CloudWatch Pengguna Amazon.

CloudWatch Sintaks kueri Log Insights

Dengan Wawasan CloudWatch Log, Anda menggunakan bahasa kueri untuk menanyakan grup log Anda. Sintaks kueri mendukung berbagai fungsi dan operasi yang menyertakan tetapi tidak terbatas pada fungsi umum, operasi aritmatika dan perbandingan, dan ekspresi reguler.

Untuk membuat kueri yang berisi beberapa perintah, pisahkan perintah dengan karakter pipa (|).

Untuk membuat kueri yang berisi komentar, matikan komentar dengan karakter hash (#).

> **Note**
>
> CloudWatch Logs Insights secara otomatis menemukan bidang untuk jenis log yang berbeda dan menghasilkan bidang yang dimulai dengan karakter @. Untuk informasi selengkapnya tentang bidang ini, lihat Log yang didukung dan bidang yang ditemukan di Panduan CloudWatch Pengguna Amazon.

Tabel berikut menjelaskan secara singkat setiap perintah. Mengikuti tabel ini adalah deskripsi yang lebih komprehensif dari setiap perintah, dengan contoh.

> **Note**
>
> Semua perintah kueri Wawasan CloudWatch Log didukung pada grup log di kelas log Standar. Grup log di kelas log Akses Jarang mendukung semua perintah kueri kecuali`pattern`,`diff`, dan`unmask`.

`display`	Menampilkan bidang atau bidang tertentu dalam hasil kueri.
`fields`	Menampilkan bidang tertentu dalam hasil kueri dan mendukung fungsi dan operasi yang dapat Anda gunakan untuk memodifikasi nilai bidang dan membuat bidang baru untuk digunakan dalam kueri Anda.

filter	Memfilter kueri untuk mengembalikan hanya peristiwa log yang cocok dengan satu atau beberapa kondisi.
pattern	Secara otomatis mengelompokkan data log Anda ke dalam pola. Pola adalah struktur teks bersama yang berulang di antara bidang log Anda. CloudWatch Logs Insights menyediakan cara bagi Anda untuk menganalisis pola yang ditemukan dalam peristiwa log Anda. Untuk informasi selengkapnya, lihat Analisis pola.
diff	Membandingkan peristiwa log yang ditemukan dalam periode waktu yang Anda minta dengan peristiwa log dari periode waktu sebelumnya dengan panjang yang sama, sehingga Anda dapat mencari tren dan mencari tahu apakah peristiwa log tertentu baru.
parse	Mengekstrak data dari bidang log untuk membuat bidang yang diekstraksi yang dapat Anda proses dalam kueri Anda. **parse**mendukung mode glob menggunakan wildcard, dan ekspresi reguler.
sort	Menampilkan peristiwa log yang dikembalikan dalam urutan ascending (`asc`) atau descending (`desc`).
stats	Hitung statistik agregat menggunakan nilai di bidang log.
limit	Menentukan jumlah maksimum peristiwa log yang Anda ingin query Anda untuk kembali. **sort**Berguna dengan mengembalikan hasil “20 teratas” atau “20 terbaru”.
dedup	Menghapus hasil duplikat berdasarkan nilai tertentu di bidang yang Anda tentukan.
unmask	Menampilkan semua konten peristiwa log yang memiliki beberapa konten yang disembunyikan karena kebijakan perlindungan data. Untuk informasi selengkapnya tentang perlindungan data di grup log, lihatMembantu melindungi data log sensitif dengan masking.
Operasi dan fungsi lainnya	CloudWatch Logs Insights juga mendukung banyak perbandingan, aritmatika, datetime, numerik, string, alamat IP, dan fungsi dan operasi umum.

Bagian berikut memberikan detail selengkapnya tentang perintah kueri Wawasan CloudWatch Log.

Topik

tampilan

Gunakan `display` untuk menampilkan bidang atau bidang tertentu dalam hasil kueri.

`display`Perintah hanya menampilkan bidang yang Anda tentukan. Jika kueri Anda berisi beberapa `display` perintah, hasil kueri hanya menampilkan bidang atau bidang yang Anda tentukan dalam `display` perintah akhir.

Contoh: Menampilkan satu bidang

Cuplikan kode menunjukkan contoh kueri yang menggunakan perintah parse untuk mengekstrak data dari `@message` untuk membuat bidang yang diekstraksi dan. `loggingType` `loggingMessage` Query mengembalikan semua peristiwa log di mana nilai-nilai untuk `loggingType` adalah ERROR. `display`hanya menampilkan nilai untuk `loggingMessage` dalam hasil query.

```
fields @message
```

```
| parse @message "[*] *" as loggingType, loggingMessage
| filter loggingType = "ERROR"
| display loggingMessage
```

> **Tip**
>
> Gunakan `display` hanya sekali dalam kueri. Jika Anda menggunakan `display` lebih dari sekali dalam kueri, hasil kueri menunjukkan bidang yang ditentukan dalam kemunculan terakhir `display` perintah yang digunakan.

ladang

Gunakan `fields` untuk menampilkan bidang tertentu dalam hasil kueri.

Jika kueri Anda berisi beberapa `fields` perintah dan tidak menyertakan `display` perintah, hasilnya akan menampilkan semua bidang yang ditentukan dalam `fields` perintah.

Contoh: Menampilkan bidang tertentu

Contoh berikut menunjukkan query yang mengembalikan 20 peristiwa log dan menampilkannya dalam urutan menurun. Nilai untuk `@timestamp` dan `@message` ditampilkan dalam hasil query.

```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
```

Gunakan `fields` sebagai gantinya`display`. ketika Anda ingin menggunakan berbagai fungsi dan operasi yang didukung oleh `fields` untuk memodifikasi nilai bidang dan membuat bidang baru yang dapat digunakan dalam kueri.

Anda dapat menggunakan *`fields`* perintah dengan kata kunci untuk membuat bidang yang diekstraksi yang menggunakan bidang dan fungsi dalam peristiwa log Anda. Misalnya, `fields ispresent as isRes` membuat bidang yang diekstraksi bernama`isRes`, dan bidang yang diekstraksi dapat digunakan di sisa kueri Anda.

filter

Gunakan `filter` untuk mendapatkan peristiwa log yang cocok dengan satu atau beberapa kondisi.

Contoh: Filter peristiwa log menggunakan satu kondisi

Cuplikan kode menunjukkan contoh kueri yang mengembalikan semua peristiwa log di mana nilainya ***range*** lebih besar dari 3000. Kueri membatasi hasil hingga 20 peristiwa log dan mengurutkan peristiwa log berdasarkan `@timestamp` dan dalam urutan menurun.

```
fields @timestamp, @message
| filter (range>3000)
| sort @timestamp desc
| limit 20
```

Contoh: Filter peristiwa log menggunakan lebih dari satu kondisi

Anda dapat menggunakan kata kunci `and` dan `or` menggabungkan lebih dari satu kondisi.

Cuplikan kode menunjukkan contoh kueri yang mengembalikan peristiwa log di mana nilai untuk lebih besar dari 3000 dan nilai untuk ***range*** sama dengan ***accountId*** 123456789012. Kueri membatasi hasil hingga 20 peristiwa log dan mengurutkan peristiwa log berdasarkan `@timestamp` dan dalam urutan menurun.

```
fields @timestamp, @message
| filter (range>3000 and accountId=123456789012)
| sort @timestamp desc
| limit 20
```

Kecocokan dan ekspresi reguler dalam perintah filter

Perintah filter mendukung penggunaan ekspresi reguler. Anda dapat menggunakan operator perbandingan berikut (=,,`!=`,<, <=>,>=) dan operator Boolean (`and`,`or`, dan`not`).

Anda dapat menggunakan kata kunci `in` untuk menguji keanggotaan set dan memeriksa elemen dalam array. Untuk memeriksa elemen dalam array, letakkan array setelahnya`in`. Anda dapat menggunakan operator Boolean `not` dengan`in`. Anda dapat membuat kueri yang digunakan `in` untuk mengembalikan peristiwa log di mana bidang cocok dengan string. Bidang harus berupa string lengkap. Misalnya, cuplikan kode berikut menunjukkan kueri yang digunakan `in` untuk mengembalikan peristiwa log di mana bidang `logGroup` adalah string lengkap. `example_group`

```
fields @timestamp, @message
| filter logGroup in ["example_group"]
```

Anda dapat menggunakan frase kata kunci `like` dan `not like` untuk mencocokkan substring. Anda dapat menggunakan operator ekspresi reguler =~ untuk mencocokkan substring. Untuk mencocokkan substring dengan `like` dan`not like`, lampirkan substring yang ingin Anda cocokkan dalam tanda kutip tunggal atau ganda. Anda dapat menggunakan pola ekspresi reguler dengan `like` dan`not like`. Untuk mencocokkan substring dengan operator ekspresi reguler, lampirkan substring yang ingin Anda cocokkan dalam garis miring maju. Contoh berikut berisi cuplikan kode yang menunjukkan bagaimana Anda dapat mencocokkan substring menggunakan perintah. `filter`

Contoh: Match substring

Contoh berikut mengembalikan peristiwa log yang `f1` berisi kata Pengecualian. Ketiga contoh tersebut peka terhadap huruf besar dan kecil.

Contoh pertama cocok dengan substring dengan`like`.

```
fields f1, f2, f3
| filter f1 like "Exception"
```

Contoh kedua cocok dengan substring dengan `like` dan pola ekspresi reguler.

```
fields f1, f2, f3
| filter f1 like /Exception/
```

Contoh ketiga cocok dengan substring dengan ekspresi reguler.

```
fields f1, f2, f3
| filter f1 =~ /Exception/
```

Contoh: Cocokkan substring dengan wildcard

Anda dapat menggunakan simbol periode (`.`) sebagai wildcard dalam ekspresi reguler untuk mencocokkan substring. Dalam contoh berikut, query mengembalikan kecocokan di mana nilai untuk `f1` dimulai dengan string`ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog./
```

Anda dapat menempatkan simbol tanda bintang setelah simbol periode (`.*`) untuk membuat kuantifier serakah yang mengembalikan kecocokan sebanyak mungkin. Misalnya, query berikut

mengembalikan kecocokan di mana nilai untuk `f1` tidak hanya dimulai dengan string`ServiceLog`, tetapi juga termasuk string`ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog.*/
```

Kemungkinan kecocokan dapat diformat seperti berikut:

- `ServiceLogSampleApiLogGroup`
- `SampleApiLogGroupServiceLog`

Contoh: Kecualikan substring dari korek api

Contoh berikut menunjukkan query yang mengembalikan peristiwa log di mana `f1` tidak mengandung kata Exception. Contohnya adalah case senstive.

```
fields f1, f2, f3
| filter f1 not like "Exception"
```

Contoh: Cocokkan substring dengan pola case-insensitive

Anda dapat mencocokkan substring yang tidak peka huruf besar/kecil dengan ekspresi `like` reguler. Tempatkan parameter berikut (? i) sebelum substring yang ingin Anda cocokkan. Contoh berikut menunjukkan query yang mengembalikan peristiwa log yang `f1` berisi kata Pengecualian atau pengecualian.

```
fields f1, f2, f3
| filter f1 like /(?i)Exception/
```

pola

Gunakan `pattern` untuk secara otomatis mengelompokkan data log Anda ke dalam pola.

Pola adalah struktur teks bersama yang berulang di antara bidang log Anda. Anda dapat menggunakan `pattern` untuk memunculkan tren yang muncul, memantau kesalahan yang diketahui, dan mengidentifikasi jalur log yang sering terjadi atau berbiaya tinggi. CloudWatch Logs Insights juga menyediakan pengalaman konsol yang dapat Anda gunakan untuk menemukan dan

menganalisis pola lebih lanjut dalam peristiwa log Anda. Untuk informasi selengkapnya, lihat Analisis pola.

Karena `pattern` perintah secara otomatis mengidentifikasi pola umum, Anda dapat menggunakannya sebagai titik awal untuk mencari dan menganalisis log Anda. Anda juga dapat menggabungkan `pattern` dengan `filter`, `parse`, atau `sort` perintah untuk mengidentifikasi pola dalam kueri yang lebih disempurnakan.

Masukan Perintah Pola

`pattern`Perintah mengharapkan salah satu input berikut: `@message` bidang, bidang yang diekstraksi yang dibuat menggunakan `parse` perintah, atau string yang dimanipulasi menggunakan satu atau beberapa fungsi String.

Output Perintah Pola

`pattern`Perintah menghasilkan output berikut:

- `@pattern`: Struktur teks bersama yang berulang di antara bidang peristiwa log Anda. Bidang yang bervariasi dalam suatu pola, seperti ID permintaan atau stempel waktu, diwakili oleh. `<*>` Misalnya, `[INFO] Request time: <*> ms` adalah output potensial untuk pesan log`[INFO] Request time: 327 ms`.
- `@ratio`: Rasio peristiwa log dari periode waktu yang dipilih dan grup log tertentu yang cocok dengan pola yang diidentifikasi. Misalnya, jika setengah dari peristiwa log dalam grup log yang dipilih dan periode waktu cocok dengan pola, `@ratio` kembali `0.50`
- `@sampleCount`: Hitungan jumlah peristiwa log dari periode waktu yang dipilih dan grup log tertentu yang cocok dengan pola yang diidentifikasi.
- `@severityLabel`: Tingkat keparahan atau tingkat log, yang menunjukkan jenis informasi yang terkandung dalam log. Contohnya: `Error`, `Warning`, `Info`, atau `Debug`.

Contoh

Perintah berikut mengidentifikasi log dengan struktur serupa dalam grup log tertentu selama rentang waktu yang dipilih, mengelompokkannya berdasarkan pola dan hitungan

```
pattern @message
```

`pattern`Perintah dapat digunakan dalam kombinasi dengan `filter` perintah

```
filter @message like /ERROR/
| pattern @message
```

`pattern`Perintah dapat digunakan dengan `sort` perintah `parse` dan

```
filter @message like /ERROR/
| parse @message 'Failed to do: *' as cause
| pattern cause
| sort @sampleCount asc
```

diff

Membandingkan peristiwa log yang ditemukan dalam periode waktu yang Anda minta dengan peristiwa log dari periode waktu sebelumnya dengan panjang yang sama. Dengan cara ini, Anda dapat mencari tren dan menemukan apakah peristiwa log tertentu baru.

Tambahkan pengubah ke `diff` perintah untuk menentukan periode waktu yang ingin Anda bandingkan dengan:

- `diff`membandingkan peristiwa log dalam rentang waktu yang dipilih saat ini dengan peristiwa log dari rentang waktu sebelumnya.
- `diff previousDay`membandingkan peristiwa log dalam rentang waktu yang dipilih saat ini dengan peristiwa log dari waktu yang sama pada hari sebelumnya.
- `diff previousWeek`membandingkan peristiwa log dalam rentang waktu yang dipilih saat ini dengan peristiwa log dari waktu yang sama minggu sebelumnya.
- `diff previousMonth`membandingkan peristiwa log dalam rentang waktu yang dipilih saat ini dengan peristiwa log dari waktu yang sama pada bulan sebelumnya.

Untuk informasi selengkapnya, lihat Bandingkan (diff) dengan rentang waktu sebelumnya.

mengurai

Gunakan `parse` untuk mengekstrak data dari bidang log dan membuat bidang yang diekstraksi yang dapat Anda proses dalam kueri Anda. **`parse`**mendukung mode glob menggunakan wildcard, dan ekspresi reguler. Untuk informasi tentang sintaks ekspresi reguler, lihatSintaks ekspresi reguler (regex) yang didukung.

Anda dapat mengurai bidang JSON bersarang dengan ekspresi reguler.

Contoh: Mengurai bidang JSON bersarang

Cuplikan kode menunjukkan cara mengurai peristiwa log JSON yang telah diratakan selama konsumsi.

```
{'fieldsA': 'logs', 'fieldsB': [{'fA': 'a1'}, {'fA': 'a2'}]}
```

Cuplikan kode menunjukkan kueri dengan ekspresi reguler yang mengekstrak nilai `fieldsB` untuk `fieldsA` dan membuat bidang yang diekstraksi dan. `fld` `array`

```
parse @message "'fieldsA': '*', 'fieldsB': ['*']" as fld, array
```

Dinamakan menangkap kelompok

Bila Anda menggunakan **`parse`**dengan ekspresi reguler, Anda dapat menggunakan grup penangkap bernama untuk menangkap pola ke dalam bidang. Sintaksnya adalah `parse @message (?<Name>pattern).`

Contoh berikut menggunakan grup menangkap pada log aliran VPC untuk mengekstrak ENI ke dalam bidang bernama. `NetworkInterface`

```
parse @message /(?<NetworkInterface>eni-.*?) / display @timestamp, NetworkInterface
```

> **Note**
>
> Peristiwa log JSON diratakan selama konsumsi. Saat ini, mengurai bidang JSON bersarang dengan ekspresi glob tidak didukung. Anda hanya dapat mengurai peristiwa log JSON yang menyertakan tidak lebih dari 200 bidang peristiwa log. Saat Anda mengurai bidang JSON bersarang, Anda harus memformat ekspresi reguler dalam kueri agar sesuai dengan format peristiwa log JSON Anda.

Contoh perintah parse

Gunakan ekspresi glob untuk mengekstrak bidang**`@user`**,**`@method`**, dan **`@latency`** dari bidang log **`@message`** dan kembalikan latensi rata-rata untuk setiap kombinasi unik dan. **`@method`** **`@user`**

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Gunakan ekspresi reguler untuk mengekstrak bidang**@user2**,**@method2**, dan **@latency2** dari bidang log **@message** dan kembalikan latensi rata-rata untuk setiap kombinasi unik **@method2** dan**@user2**.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Mengekstrak bidang**loggingTime**, **loggingType** dan**loggingMessage**, memfilter ke log peristiwa yang berisi **ERROR** atau **INFO** string, dan kemudian hanya menampilkan **loggingMessage** dan **loggingType** bidang untuk peristiwa yang berisi **ERROR** string.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

menyortir

Gunakan `sort` untuk menampilkan peristiwa log dalam urutan ascending (`asc`) atau descending (`desc`) dengan bidang tertentu. Anda dapat menggunakan ini dengan `limit` perintah untuk membuat kueri “N atas” atau “N bawah”.

Algoritma penyortiran adalah versi terbaru dari penyortiran alami. Jika Anda mengurutkan dalam urutan menaik, logika berikut digunakan.

- Semua nilai non-angka datang sebelum semua nilai angka. Nilai angka adalah nilai yang hanya mencakup angka, bukan campuran angka dan karakter lainnya.
- Untuk nilai non-angka, algoritme mengelompokkan karakter numerik berurutan dan karakter alfabet berurutan ke dalam potongan terpisah untuk perbandingan. Ini memesan bagian non-numerik dengan nilai Unicode mereka, dan mengurutkan bagian numerik berdasarkan panjangnya terlebih dahulu dan kemudian dengan nilai numeriknya.

Untuk informasi selengkapnya tentang urutan Unicode, lihat Daftar karakter Unicode.

Misalnya, berikut ini adalah hasil dari semacam dalam urutan menaik.

```
!: >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> sorted by unicode order
#
*%04
0# >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Alphanumeric starting with numbers
5A
111A   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  Starts with more digits than 5A, so it sorted
 to be later than 5A
2345_
@ >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> 2345 is compared with @ in the unicode order,
@_
A >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Values starting with letters
A9876fghj
a12345hfh
0 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Number values
01
1
2
3
```

Jika Anda mengurutkan dalam urutan menurun, hasil urutannya adalah sebaliknya.

Misalnya, kueri berikut untuk log aliran VPC Amazon menemukan 15 transfer paket teratas di seluruh host.

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

statistik

Gunakan `stats` untuk membuat visualisasi data log Anda seperti diagram batang, diagram garis, dan bagan area bertumpuk. Ini membantu Anda mengidentifikasi pola dalam data log Anda dengan lebih efisien. CloudWatch Log Insights menghasilkan visualisasi untuk kueri yang menggunakan `stats` fungsi dan satu atau beberapa fungsi agregasi.

Misalnya, kueri berikut dalam grup log Route 53 mengembalikan visualisasi yang menunjukkan distribusi catatan Route 53 per jam, berdasarkan jenis kueri.

```
stats count(*) by queryType, bin(1h)
```

Semua kueri tersebut dapat menghasilkan diagram batang. Jika kueri Anda menggunakan fungsi `bin()` untuk mengelompokkan data dengan satu bidang dari waktu ke waktu, Anda juga dapat melihat diagram garis dan diagram area bertumpuk.

Satuan waktu dan singkatan berikut didukung dengan `bin` fungsi tersebut. Untuk semua unit dan singkatan yang menyertakan lebih dari satu karakter, menambahkan s ke pluralisasi didukung. Jadi keduanya `hr` dan `hrs` bekerja untuk menentukan jam.

- `millisecond` `ms` `msec`
- `second` `s` `sec`
- `minute` `m` `min`
- `hour` `h` `hr`
- `day` `d`
- `week` `w`
- `month` `mo` `mon`
- `quarter` `q` `qtr`
- `year` `y` `yr`

Topik

- Visualisasikan data deret waktu
- Visualisasikan data log yang dikelompokkan berdasarkan bidang
- Gunakan beberapa perintah statistik dalam satu kueri
- Fungsi untuk digunakan dengan statistik

Visualisasikan data deret waktu

Visualisasi deret waktu dapat digunakan dengan kueri yang memiliki karakteristik berikut:

- Kueri berisi satu atau beberapa fungsi agregasi. Untuk informasi selengkapnya, lihat Aggregation Functions in the Stats Command.
- Kueri menggunakan fungsi `bin()` untuk mengelompokkan data dengan satu bidang.

Kueri-kueri ini dapat menghasilkan diagram garis, diagram area bertumpuk, diagram batang, dan diagram lingkaran.

Contoh

Untuk tutorial lengkap, lihat the section called “Tutorial: Jalankan kueri yang menghasilkan visualisasi deret waktu”.

Berikut adalah contoh kueri lain yang dapat digunakan untuk visualisasi deret waktu.

Kueri berikut menghasilkan visualisasi nilai rata-rata bidang `myfield1`, dengan titik data yang dibuat setiap lima menit. Setiap titik data adalah agregasi dari rata-rata nilai `myfield1` dari log lima menit sebelumnya.

```
stats avg(myfield1) by bin(5m)
```

Kueri berikut menghasilkan visualisasi dari tiga nilai berdasarkan pada bidang-bidang yang berbeda, dengan titik data yang dibuat setiap lima menit. Visualisasi dihasilkan karena kueri berisi fungsi agregat dan menggunakan `bin()` sebagai bidang pengelompokan.

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

Bagan garis dan batasan bagan area bertumpuk

Pertanyaan yang membuat agregat informasi entri log, tetapi tidak menggunakan fungsi `bin()` dapat menghasilkan diagram batang. Namun, kueri tidak dapat menghasilkan diagram garis atau diagram area bertumpuk. Untuk informasi selengkapnya tentang tipe kueri ini, lihat the section called “Visualisasikan data log yang dikelompokkan berdasarkan bidang”.

Visualisasikan data log yang dikelompokkan berdasarkan bidang

Anda dapat menghasilkan diagram batang untuk kueri yang menggunakan fungsi `stats` dan satu atau beberapa fungsi agregasi. Untuk informasi selengkapnya, lihat Aggregation Functions in the Stats Command.

Untuk melihat visualisasi, jalankan kueri Anda. Lalu pilih tab Visualization (Visualisasi), pilih panah di sebelah Line (Garis), dan pilih Bar (Batang). Visualisasi dibatasi hingga 100 batang dalam diagram batang.

Contoh

Untuk tutorial lengkap, lihat the section called “Tutorial: Jalankan kueri yang menghasilkan visualisasi yang dikelompokkan berdasarkan bidang log”. Paragraf berikut mencakup lebih banyak contoh kueri untuk visualisasi berdasarkan bidang.

Kueri log alur VPC berikut menemukan jumlah rata-rata byte yang ditransfer per sesi untuk setiap alamat tujuan.

```
stats avg(bytes) by dstAddr
```

Anda juga dapat menghasilkan diagram yang mencakup lebih dari satu batang untuk setiap nilai yang dihasilkan. Misalnya, kueri log alur VPC berikut menemukan jumlah rata-rata dan maksimum byte yang ditransfer per sesi untuk setiap alamat tujuan.

```
stats avg(bytes), max(bytes) by dstAddr
```

Kueri berikut menemukan jumlah log kueri Amazon Route 53 untuk setiap jenis kueri.

```
stats count(*) by queryType
```

Gunakan beberapa perintah statistik dalam satu kueri

Anda dapat menggunakan sebanyak dua `stats` perintah dalam satu kueri. Ini memungkinkan Anda untuk melakukan agregasi tambahan pada output agregasi pertama.

Contoh: Query dengan dua **`stats`** perintah

Misalnya, kueri berikut pertama-tama menemukan total volume lalu lintas di tempat sampah 5 menit, kemudian menghitung volume lalu lintas tertinggi, terendah, dan rata-rata di antara tempat sampah 5 menit tersebut.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length)/1024/1024 as logs_mb BY bin(5m)
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb
```

Contoh: Menggabungkan beberapa perintah statistik dengan fungsi lain seperti**`filter`**,, **`fields`** **`bin`**

Anda dapat menggabungkan dua `stats` perintah dengan perintah lain seperti `filter` dan `fields` dalam satu kueri. Misalnya, kueri berikut menemukan jumlah alamat IP yang berbeda dalam sesi dan menemukan jumlah sesi berdasarkan platform klien, memfilter alamat IP tersebut, dan akhirnya menemukan rata-rata permintaan sesi per platform klien.

```
STATS count_distinct(client_ip) AS session_ips,
      count(*) AS requests BY session_id, client_platform
| FILTER session_ips > 1
| STATS count(*) AS multiple_ip_sessions,
        sum(requests) / count(*) AS avg_session_requests BY client_platform
```

Anda dapat menggunakan `bin` dan `dateceil` berfungsi dalam kueri dengan beberapa `stats` perintah. Misalnya, kueri berikut pertama-tama menggabungkan pesan menjadi blok 5 menit, kemudian menggabungkan blok 5 menit tersebut menjadi blok 10 menit dan menghitung volume lalu lintas tertinggi, terendah, dan rata-rata dalam setiap blok 10 menit.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length) / 1024 / 1024 AS logs_mb BY BIN(5m) as @t
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb BY dateceil(@t, 10m)
```

Catatan dan batasan

Kueri dapat memiliki maksimal dua `stats` perintah. Kuota ini tidak dapat diubah.

Jika Anda menggunakan `limit` perintah `sort` atau, itu harus muncul setelah `stats` perintah kedua. Jika sebelum `stats` perintah kedua, kueri tidak valid.

Ketika kueri memiliki dua `stats` perintah, sebagian hasil dari kueri tidak mulai ditampilkan sampai `stats` agregasi pertama selesai.

Dalam `stats` perintah kedua dalam satu kueri, Anda hanya dapat merujuk ke bidang yang didefinisikan dalam `stats` perintah pertama. Misalnya, kueri berikut tidak valid karena `@message` bidang tidak akan tersedia setelah `stats` agregasi pertama.

```
FIELDS @message
| STATS SUM(Fault) by Operation
# You can only reference `SUM(Fault)` or Operation at this point
| STATS MAX(strlen(@message)) AS MaxMessageSize # Invalid reference to @message
```

Bidang apa pun yang Anda referensikan setelah `stats` perintah pertama harus didefinisikan dalam `stats` perintah pertama itu.

```
STATS sum(x) as sum_x by y, z
```

```
| STATS max(sum_x) as max_x by z
# You can only reference `max(sum_x)`, max_x or z at this point
```

Important

`bin`Fungsi selalu secara implisit menggunakan bidang. `@timestamp` Ini berarti bahwa Anda tidak dapat menggunakan `bin` dalam `stats` perintah kedua tanpa menggunakan `stats` perintah pertama untuk menyebarkan `timestamp` bidang. Misalnya, kueri berikut tidak valid.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes BY @logStream
 | STATS avg(ingested_bytes) BY bin(5m) # Invalid reference to @timestamp field
```

Sebagai gantinya, tentukan `@timestamp` bidang di `stats` perintah pertama, dan kemudian Anda dapat menggunakannya dengan `dateceil` `stats` perintah kedua seperti pada contoh berikut.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes, max(@timestamp) as @t BY
 @logStream
 | STATS avg(ingested_bytes) BY dateceil(@t, 5m)
```

Fungsi untuk digunakan dengan statistik

CloudWatch Logs Insights mendukung fungsi agregasi statistik dan fungsi non-agregasi statistik.

Gunakan fungsi statsaggregation dalam `stats` perintah dan sebagai argumen untuk fungsi lainnya.

Fungsi	Tipe hasil	Deskripsi
`avg(fieldName: NumericLogField)`	nomor	Rata-rata nilai di bidang yang ditentukan.

`count()`
 	
`count(fieldName: LogField)`
	nomor	Menghitung log acara. `count()` (atau `count(*)`) menghitung semua kejadian yang dikembalikan oleh kueri, sementara `count(fieldName)` menghitung semua

Fungsi	Tipe hasil	Deskripsi
		catatan yang menyertakan nama bidang yang ditentukan.
`count_distinct(fieldName: LogField)`	nomor	Mengembalikan jumlah nilai unik untuk bidang. Jika bidang memiliki kardinalitas yang sangat tinggi (mengandung banyak nilai unik), nilai yang dikembalikan oleh `count_distinct` hanyalah sebuah perkiraan.
`max(fieldName: LogField)`	LogFieldValue	Nilai maksimum untuk bidang log ini dalam log yang dikueri.
`min(fieldName: LogField)`	LogFieldValue	Nilai minimum untuk bidang log ini dalam log yang dikueri.
`pct(fieldName: LogFieldValue, percent: number)`	LogFieldValue	Persentil menunjukkan posisi relatif dari nilai dalam rangkaian data. Misalnya, `pct(@duration, 95)` mengembalikan nilai `@duration` di mana 95 persen dari nilai `@duration` lebih rendah dari nilai ini, dan 5 persen lebih tinggi dari nilai ini.
`stddev(fieldName: NumericLogField)`	nomor	Standar deviasi di bidang yang ditentukan.
`sum(fieldName: NumericLogField)`	nomor	Jumlah nilai di bidang yang ditentukan.

Statistik fungsi non-agregasi

Gunakan fungsi non-agregasi dalam `stats` perintah dan sebagai argumen untuk fungsi lainnya.

Fungsi	Tipe hasil	Deskripsi
`earliest(fieldName: LogField)`	LogField	Mengembalikan nilai `fieldName` dari log acara yang memiliki stempel waktu paling awal dalam log yang dikueri.
`latest(fieldName: LogField)`	LogField	Mengembalikan nilai `fieldName` dari log acara yang memiliki stempel waktu paling akhir dalam log yang dikueri.
`sortsFirst(fieldName: LogField)`	LogField	Mengembalikan nilai `fieldName` yang ada di urutan pertama dalam log yang dikueri.
`sortsLast(fieldName: LogField)`	LogField	Mengembalikan nilai `fieldName` yang ada di urutan terakhir dalam log yang dikueri.

batasan

Gunakan `limit` untuk menentukan jumlah peristiwa log yang Anda ingin kueri Anda kembalikan.

Misalnya, contoh berikut hanya mengembalikan 25 peristiwa log terbaru

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

dedup

Gunakan `dedup` untuk menghapus hasil duplikat berdasarkan nilai tertentu di bidang yang Anda tentukan. Anda dapat menggunakan `dedup` dengan satu atau lebih bidang. Jika Anda menentukan satu bidang dengan`dedup`, hanya satu peristiwa log yang dikembalikan untuk setiap nilai unik bidang itu. Jika Anda menentukan beberapa bidang, maka satu peristiwa log dikembalikan untuk setiap kombinasi nilai unik untuk bidang tersebut.

Duplikat dibuang berdasarkan urutan pengurutan, dengan hanya hasil pertama dalam urutan pengurutan yang disimpan. Kami menyarankan Anda mengurutkan hasil Anda sebelum memasukkannya melalui `dedup` perintah. Jika hasilnya tidak diurutkan sebelum dijalankan`dedup`, maka urutan urutan menurun default yang digunakan `@timestamp` digunakan.

Nilai nol tidak dianggap duplikat untuk evaluasi. Peristiwa log dengan nilai null untuk salah satu bidang tertentu dipertahankan. Untuk menghilangkan bidang dengan nilai nol, gunakan **`filter`**menggunakan `isPresent(field)` fungsi.

Satu-satunya perintah query yang dapat Anda gunakan dalam kueri setelah `dedup` perintah adalah`limit`.

Contoh: Lihat hanya peristiwa log terbaru untuk setiap nilai unik bidang bernama **`server`**

Contoh berikut menampilkan`timestamp`,`server`,`severity`, dan `message` bidang hanya untuk acara terbaru untuk setiap nilai unik`server`.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server
```

Untuk lebih banyak contoh kueri Wawasan CloudWatch Log, lihat. Kueri umum

membuka kedok

Gunakan `unmask` untuk menampilkan semua konten peristiwa log yang memiliki beberapa konten yang disembunyikan karena kebijakan perlindungan data. Untuk menggunakan perintah ini, Anda harus memiliki `logs:Unmask` izin.

Untuk informasi selengkapnya tentang perlindungan data di grup log, lihatMembantu melindungi data log sensitif dengan masking.

Boolean, perbandingan, numerik, datetime, dan fungsi lainnya

CloudWatch Log Insights mendukung banyak operasi dan fungsi lain dalam kueri, seperti yang dijelaskan di bagian berikut.

Topik

- Operator aritmatika
- Operator Boolean
- Operator perbandingan
- Operator numerik
- Fungsi datetime
- Fungsi umum

- Fungsi string alamat IP
- Fungsi string

Operator aritmatika

Operator aritmatika menerima tipe data numerik sebagai argumen dan mengembalikan hasil numerik. Gunakan operator aritmatika dalam `filter` dan `fields` perintah dan sebagai argumen untuk fungsi lainnya.

Operasi	Deskripsi
`a + b`	Penambahan
`a - b`	Pengurangan
`a * b`	Perkalian
`a / b`	Pembagian
`a ^ b`	Eksponensiasi (pengembalian) `2 ^ 3` `8`
`a % b`	Sisa atau modulus (pengembalian) `10 % 3` `1`

Operator Boolean

Gunakan operator Boolean**`and`**,**`or`**, dan**`not`**.

> **Note**
>
> Gunakan operator Boolean hanya dalam fungsi yang mengembalikan nilai TRUE atau FALSE.

Operator perbandingan

Operator perbandingan menerima semua tipe data sebagai argumen dan mengembalikan hasil Boolean. Gunakan operasi perbandingan dalam `filter` perintah dan sebagai argumen untuk fungsi lainnya.

Operator	Deskripsi
=	Sama
!=	Tidak sama
<	Kurang dari
>	Lebih besar dari
<=	Kurang dari atau sama dengan
>=	Lebih besar dari atau sama dengan

Operator numerik

Operasi numerik menerima tipe data numerik sebagai argumen dan mengembalikan hasil numerik. Gunakan operasi numerik dalam `filter` dan `fields` perintah dan sebagai argumen untuk fungsi lainnya.

Operasi	Tipe Hasil	Deskripsi
`abs(a: number)`	number	Nilai absolut
`ceil(a: number)`	number	Bulat ke langit-langit (bilangan bulat terkecil yang lebih besar dari nilai) a
`floor(a: number)`	number	Bulat ke lantai (bilangan bulat terbesar yang lebih kecil dari nilai) a
`greatest(a: number, ...numbers: number[])`	number	Mengembalikan nilai terbesar

Operasi	Tipe Hasil	Deskripsi
`least(a: number, ...numbers: number[])`	number	Mengembalikan nilai terkecil
`log(a: number)`	number	Log alami
`sqrt(a: number)`	number	Akar kuadrat

Fungsi datetime

Fungsi Datetime

Gunakan fungsi datetime dalam `fields` dan `filter` perintah dan sebagai argumen untuk fungsi lainnya. Gunakan fungsi ini untuk membuat bucket waktu untuk kueri dengan fungsi agregat. Gunakan periode waktu yang terdiri dari angka dan salah satu dari yang berikut:

- `ms`untuk milidetik
- `s`selama beberapa detik
- `m`selama beberapa menit
- hselama berjam-jam

Misalnya, `10m` adalah 10 menit, dan `1h` 1 jam.

> **Note**
>
> Gunakan unit waktu yang paling tepat untuk fungsi datetime Anda. CloudWatch Log membatasi permintaan Anda sesuai dengan satuan waktu yang Anda pilih. Misalnya, ini membatasi 60 sebagai nilai maksimum untuk setiap permintaan yang menggunakan`s`. Jadi, jika Anda menentukan`bin(300s)`, CloudWatch Log sebenarnya mengimplementasikan ini sebagai 60 detik, karena 60 adalah jumlah detik dalam satu menit sehingga CloudWatch Log tidak akan menggunakan angka yang lebih tinggi dari 60 dengan`s`. Untuk membuat ember 5 menit, gunakan `bin(5m)` sebagai gantinya.
> Tutup untuk `ms` adalah 1000, tutup untuk `s` dan `m` 60, dan tutupnya h adalah 24.

Tabel berikut berisi daftar fungsi datetime yang berbeda yang dapat Anda gunakan dalam perintah query. Tabel mencantumkan jenis hasil setiap fungsi dan berisi deskripsi dari setiap fungsi.

> **Tip**
>
> Saat Anda membuat perintah kueri, Anda dapat menggunakan pemilih interval waktu untuk memilih periode waktu yang ingin Anda kueri. Misalnya, Anda dapat mengatur periode waktu antara interval 5 dan 30 menit; interval 1, 3, dan 12 jam; atau kerangka waktu khusus. Anda juga dapat mengatur periode waktu antara tanggal tertentu.

Fungsi	Tipe hasil	Deskripsi

`bin(period: Period)`	Stempel Waktu	Membulatkan nilai `@timestamp` ke periode waktu tertentu dan kemudian memotong. Misalnya, `bin(5m)` bulatkan nilai `@timestamp` ke 5 menit terdekat.
		
		Anda dapat menggunakan ini untuk mengelompokkan beberapa entri log bersama-sama dalam kueri. Contoh berikut mengembalikan jumlah pengecualian per jam:
		
		`filter @message like /Exception/`
		`  | stats count(*) as exceptionCount`
		`by bin(1h)`
		`  | sort exceptionCount desc`
		
		Satuan waktu dan singkatan berikut didukung dengan `bin` fungsi tersebut. Untuk semua unit dan singkatan yang menyertakan lebih dari satu karakter, menambahkan `s` ke pluralisasi didukung. Jadi keduanya `hr` dan `hrs` bekerja untuk menentukan jam.
		
		• `millisecond ms msec`
		• `second s sec`

Fungsi	Tipe hasil	Deskripsi
		• `minute m min` • `hour h hr` • `day d` • `week w` • `month mo mon` • `quarter q qtr` • `year y yr`
`datefloor(timestamp: Timestamp, period: Period)`	Stempel Waktu	Memotong stempel waktu ke periode tertentu. Misalnya, `datefloor(@timestamp, 1h)` memotong semua nilai `@timestamp` ke bagian bawah jam.
`dateceil(timestamp : Timestamp, period: Period)`	Stempel waktu	Membulatkan stempel waktu ke periode tertentu dan kemudian memotong. Misalnya, `dateceil(@timestamp, 1h)` memotong semua nilai `@timestamp` ke bagian atas jam.
`fromMillis(fieldNa me: number)`	Stempel waktu	Menafsirkan bidang input sebagai jumlah milidetik sejak jangka waktu Unix dan mengubahnya menjadi stempel waktu.
`toMillis(fieldName: Timestamp)`	nomor	Mengonversi stempel waktu yang ditemukan di bidang bernama menjadi angka yang mewakili milidetik sejak jangka waktu Unix. Misalnya, `toMillis(@timestamp)` mengubah stempel waktu `2022-01-14T13:18:0 31.000-08:00` menjadi. `1642195111000`

> **Note**
>
> Saat ini, CloudWatch Logs Insights tidak mendukung pemfilteran log dengan stempel waktu yang dapat dibaca manusia.

Fungsi umum

Fungsi umum

Gunakan fungsi umum dalam `fields` dan `filter` perintah dan sebagai argumen untuk fungsi lainnya.

Fungsi	Tipe hasil	Deskripsi
`ispresent(fieldName: LogField)`	Boolean	Mengembalikan `true` jika bidang ada
`coalesce(fieldName: LogField, ...fieldNames: LogField[])`	LogField	Mengembalikan nilai non-null pertama dari daftar

Fungsi string alamat IP

Fungsi string alamat IP

Gunakan fungsi string alamat IP dalam `filter` dan `fields` perintah dan sebagai argumen untuk fungsi lainnya.

Fungsi	Tipe hasil	Deskripsi
`isValidIp(fieldName: string)`	boolean	Mengembalikan `true` jika bidang adalah alamat IPv4 atau IPv6 yang valid.
`isValidIpV4(fieldName: string)`	boolean	Mengembalikan `true` jika bidang adalah alamat IPv4 yang valid.
`isValidIpV6(fieldName: string)`	boolean	Mengembalikan `true` jika bidang adalah alamat IPv6 yang valid.
`isIpInSubnet(fieldName: string, subnet: string)`	boolean	Mengembalikan `true` jika bidang adalah alamat IPv4 atau IPv6 yang valid dengan subnet v4 atau v6 yang ditentukan. Saat Anda menentukan subnet, gunakan notasi CIDR

Fungsi	Tipe hasil	Deskripsi
		seperti `192.0.2.0/24` atau`2001:db8::/32`, di mana `192.0.2.0` atau `2001:db8::` merupakan awal dari blok CIDR.
`isIpv4InSubnet(fieldName: string, subnet: string)`	boolean	Mengembalikan `true` jika bidang adalah alamat IPv4 yang valid dalam subnet v4 yang ditentukan. Saat Anda menentukan subnet, gunakan notasi CIDR seperti `192.0.2.0/24` di `192.0.2.0` mana awal blok CIDR..
`isIpv6InSubnet(fieldName: string, subnet: string)`	boolean	Mengembalikan `true` jika bidang adalah alamat IPv6 yang valid dalam subnet v6 yang ditentukan. Saat Anda menentukan subnet, gunakan notasi CIDR seperti `2001:db8::/32` di `2001:db8::` mana awal blok CIDR.

Fungsi string

Fungsi string

Gunakan fungsi string dalam `fields` dan `filter` perintah dan sebagai argumen untuk fungsi lainnya.

Fungsi	Tipe hasil	Deskripsi
`isempty(fieldName: string)`	Jumlah	Mengembalikan 1 jika bidang tidak ada atau string kosong.
`isblank(fieldName: string)`	Jumlah	Mengembalikan 1 jika bidang tidak ada, string kosong, atau hanya berisi spasi.
`concat(str: string, ...strings: string[])`	string	Menyatukan string.

Fungsi	Tipe hasil	Deskripsi

`ltrim(str: string)`
 	
`ltrim(str: string, trimChars: string)`
	string	Jika fungsi tidak memiliki argumen kedua, ia menghapus spasi putih dari kiri string. Jika fungsi memiliki argumen string kedua, itu tidak menghapus spasi putih. Sebaliknya, ia menghapus karakter `trimChars` dari kiri`str`. Misalnya, `ltrim("xyZxyfooxyZ","xyZ")` mengembalikan `"fooxyZ"`.

`rtrim(str: string)`
 	
`rtrim(str: string, trimChars: string)`
	string	Jika fungsi tidak memiliki argumen kedua, ia menghapus spasi putih dari kanan string. Jika fungsi memiliki argumen string kedua, itu tidak menghapus spasi putih. Sebaliknya, ia menghapus karakter `trimChars` dari dari kanan`str`. Misalnya, `rtrim("xyZfooxyxyZ","xyZ")` mengembalikan `"xyZfoo"`.

Fungsi	Tipe hasil	Deskripsi

`trim(str: string)`
 	
`trim(str: string, trimChars: string)`
	string	Jika fungsi tidak memiliki argumen kedua, ia menghapus spasi putih dari kedua ujung string. Jika fungsi memiliki argumen string kedua, itu tidak menghapus spasi putih. Sebaliknya, ia menghapus karakter `trimChars` dari kedua sisi`str`. Misalnya, `trim("xyZxyfooxyxyZ","xyZ")` mengembalikan `"foo"`.

`strlen(str: string)`	nomor	Mengembalikan panjang string dalam poin kode Unicode.
`toupper(str: string)`	string	Mengonversi string menjadi huruf besar.
`tolower(str: string)`	string	Mengonversi string menjadi huruf kecil.

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	string	Mengembalikan substring dari indeks yang ditentukan oleh argumen angka ke akhir string. Jika fungsi memiliki argumen angka kedua, itu berisi panjang substring yang akan diambil. Misalnya, `substr("xyZfooxyZ",3, 3)` mengembalikan `"foo"`.

Fungsi	Tipe hasil	Deskripsi
`replace(fieldName: string, searchValue: string, replaceValue: string)`	string	Mengganti semua `searchValue` dalam `fieldName: string`dengan `replaceValue` . Misalnya, fungsi `replace(logGroup,"smoke_test","Smoke")` mencari peristiwa log di mana bidang `logGroup` berisi nilai string `smoke_test` dan mengganti kan nilai dengan string. `Smoke`
`strcontains(str: string, searchValue: string)`	number	Mengembalikan 1 jika `str` berisi `searchValue` dan 0 sebaliknya.

Bidang yang berisi karakter khusus

Jika bidang berisi karakter non-alfanumerik selain `@` simbol atau periode (`.`), Anda harus mengelilingi bidang dengan karakter backtick (). ` Misalnya, bidang log `foo-bar` harus diapit backticks (`` `foo-bar` ``) karena berisi karakter non-alfanumerik, tanda hubung (). `-`

Gunakan alias dan komentar dalam kueri

Buat kueri yang berisi alias. Gunakan alias untuk mengganti nama bidang log atau saat mengekstrak nilai ke dalam bidang. Gunakan kata kunci `as` untuk memberikan bidang log atau menghasilkan alias. Anda dapat menggunakan lebih dari satu alias dalam kueri. Anda dapat menggunakan alias dalam perintah berikut:

- `fields`
- `parse`
- `sort`
- `stats`

Contoh berikut menunjukkan cara membuat kueri yang berisi alias.

Contoh

Query berisi alias dalam `fields` perintah.

```
fields @timestamp, @message, accountId as ID
| sort @timestamp desc
| limit 20
```

Query mengembalikan nilai-nilai untuk bidang`@timestamp`,`@message`, dan`accountId`. Hasilnya diurutkan dalam urutan menurun dan dibatasi hingga 20. Nilai untuk `accountId` tercantum di bawah alias`ID`.

Contoh

Kueri berisi alias dalam `stats` perintah `sort` dan.

```
stats count(*) by duration as time
| sort time desc
```

Kueri menghitung berapa kali bidang `duration` terjadi di grup log dan mengurutkan hasil dalam urutan menurun. Nilai untuk `duration` tercantum di bawah alias`time`.

Gunakan komentar

CloudWatch Log Insights mendukung komentar dalam kueri. Gunakan karakter hash (#) untuk memicu komentar. Anda dapat menggunakan komentar untuk mengabaikan baris dalam kueri atau kueri dokumen.

Contoh: Query

Ketika query berikut dijalankan, baris kedua diabaikan.

```
fields @timestamp, @message, accountId
# | filter accountId not like "7983124201998"
| sort @timestamp desc
| limit 20
```

Analisis pola

CloudWatch Logs Insights menggunakan algoritme pembelajaran mesin untuk menemukan pola saat Anda menanyakan log Anda. Pola adalah struktur teks bersama yang berulang di antara bidang log Anda. Saat melihat hasil kueri, Anda dapat memilih tab Pola untuk melihat pola yang ditemukan CloudWatch Log berdasarkan sampel hasil Anda. Atau, Anda dapat menambahkan `pattern` perintah ke kueri Anda untuk menganalisis pola di seluruh rangkaian peristiwa log yang cocok.

Pola berguna untuk menganalisis kumpulan log besar karena sejumlah besar peristiwa log sering dapat dikompresi menjadi beberapa pola.

Pertimbangkan contoh berikut dari tiga peristiwa log.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

Dalam sampel sebelumnya, ketiga peristiwa log mengikuti satu pola:

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

Bidang dalam pola disebut token. Bidang yang bervariasi dalam suatu pola, seperti ID permintaan atau stempel waktu, adalah token dinamis. Setiap token dinamis diwakili oleh `<*>` saat CloudWatch Log menampilkannya.

Contoh umum token dinamis termasuk kode kesalahan, stempel waktu, dan ID permintaan. Nilai token mewakili nilai tertentu dari token dinamis. Misalnya, jika token dinamis mewakili kode kesalahan HTTP, maka nilai token bisa jadi`501`.

Deteksi pola juga digunakan dalam detektor anomali CloudWatch Log dan fitur perbandingan. Untuk informasi selengkapnya, lihat Deteksi anomali log dan Bandingkan (diff) dengan rentang waktu sebelumnya.

Memulai dengan analisis pola

Deteksi pola dilakukan secara otomatis dalam kueri Wawasan CloudWatch Log apa pun. Kueri yang tidak menyertakan `pattern` perintah mendapatkan peristiwa log dan pola dalam hasil.

Jika Anda menyertakan `pattern` perintah dalam kueri Anda, analisis pola dilakukan pada seluruh rangkaian peristiwa log yang cocok. Ini memberi Anda hasil pola yang lebih akurat, tetapi peristiwa

log mentah tidak dikembalikan saat Anda menggunakan `pattern` perintah. Ketika kueri tidak disertakan`pattern`, hasil pola didasarkan pada 1000 peristiwa log pertama yang dikembalikan, atau pada nilai batas yang Anda gunakan dalam kueri Anda. Jika Anda menyertakan `pattern` dalam kueri, maka hasil yang ditampilkan di tab Pola berasal dari semua peristiwa log yang cocok dengan kueri.

Untuk memulai analisis pola di Wawasan CloudWatch Log

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

   Pada halaman Wawasan Log, editor kueri berisi kueri default yang menampilkan 20 peristiwa log terbaru.
3. Hapus `| limit 20` baris di kotak kueri, sehingga kueri terlihat seperti berikut:

   ```
   fields @timestamp, @message, @logStream, @log
   | sort @timestamp desc
   ```
4. Di menu drop-down Pilih grup log, pilih satu atau beberapa grup log untuk kueri.
5. (Opsional) Gunakan pemilih interval waktu untuk memilih periode waktu yang ingin Anda kueri.

   Anda dapat memilih antara interval 5 menit dan 30 menit; Interval 1 jam, 3 jam, dan 12 jam; atau kerangka waktu khusus.
6. Pilih Jalankan kueri untuk memulai kueri.

   Saat kueri selesai berjalan, tab Log menampilkan tabel peristiwa log yang dikembalikan oleh kueri. Di atas tabel adalah pesan tentang berapa banyak catatan yang cocok dengan kueri, mirip dengan Menampilkan 1000 dari 71.101 catatan yang cocok.
7. Pilih tab Patterns.
8. Tabel sekarang menampilkan pola yang ditemukan dalam kueri. Karena kueri tidak menyertakan `pattern` perintah, tab ini hanya menampilkan pola yang ditemukan di antara 1000 peristiwa log yang ditampilkan dalam tabel di tab Log.

   Untuk setiap pola, informasi berikut ditampilkan:

   - Pola, dengan setiap token dinamis ditampilkan sebagai`<*>`.
   - Hitungan Peristiwa, yang merupakan berapa kali pola muncul dalam peristiwa log yang ditanyakan. Pilih judul kolom Event count untuk mengurutkan pola berdasarkan frekuensi.

- Rasio peristiwa, yang merupakan persentase dari peristiwa log kueri yang berisi pola ini.
- Jenis Keparahan, yang akan menjadi salah satu dari yang berikut:
  - ERROR jika pola berisi kata Error.
  - PERINGATKAN jika pola berisi kata Warn tetapi tidak mengandung Error.
  - INFO jika pola tidak mengandung Warn atau Error.

  Pilih judul kolom Info keparahan untuk mengurutkan pola berdasarkan tingkat keparahan.

9. Sekarang ubah kueri. Ganti `| sort @timestamp desc` baris dalam kueri dengan `| pattern @message`, sehingga kueri lengkapnya adalah sebagai berikut:

```
fields @timestamp, @message, @logStream, @log
| pattern @message
```

10. Pilih Run query (Jalankan kueri).

    Saat kueri selesai, tidak ada hasil di tab Log. Namun, tab Patterns kemungkinan memiliki jumlah pola yang lebih besar yang terdaftar, tergantung pada jumlah total peristiwa log yang ditanyakan.

11. Terlepas dari apakah Anda termasuk `pattern` dalam kueri Anda, Anda dapat memeriksa lebih lanjut pola yang dikembalikan kueri. Untuk melakukannya, pilih ikon di kolom Inspect untuk salah satu pola.

    Panel pemeriksaan Pola muncul dan menampilkan yang berikut:

    - Pola. Pilih token dalam pola untuk menganalisis nilai token tersebut.
    - Histogram yang menunjukkan jumlah kemunculan pola selama rentang waktu yang ditanyakan. Ini dapat membantu Anda mengidentifikasi tren menarik seperti peningkatan tiba-tiba dalam terjadinya suatu pola.
    - Tab Log samples menampilkan beberapa peristiwa log yang cocok dengan pola yang dipilih.
    - Tab Nilai Token menampilkan nilai token dinamis yang dipilih, jika Anda telah memilihnya.

    > **Note**
    >
    > Maksimal 10 nilai token ditangkap untuk setiap token. Jumlah token mungkin tidak tepat. CloudWatch Log menggunakan penghitung probabilistik untuk menghasilkan jumlah token, bukan nilai absolut.

- Tab pola Terkait menampilkan pola lain yang sering terjadi mendekati waktu yang sama dengan pola yang Anda periksa. Misalnya, jika pola untuk `ERROR` pesan biasanya disertai dengan peristiwa log lain yang ditandai `INFO` dengan detail tambahan, pola itu ditampilkan di sini.

Detail tentang perintah pola

Bagian ini berisi rincian lebih lanjut tentang `pattern` perintah dan penggunaannya.

- Dalam tutorial sebelumnya, kami menghapus `sort` perintah ketika kami menambahkan `pattern` karena kueri tidak valid jika menyertakan `pattern` perintah setelah `sort` perintah. Adalah sah untuk memiliki `pattern` sebelum `asort`.

  Untuk detail selengkapnya tentang `pattern` sintaks, lihatpola .
- Bila Anda menggunakan `pattern` dalam query, `@message` harus menjadi salah satu bidang yang dipilih dalam `pattern` perintah.
- Anda dapat menyertakan `filter` perintah sebelum `pattern` perintah untuk menyebabkan hanya kumpulan peristiwa log yang difilter untuk digunakan sebagai masukan untuk analisis pola.
- Untuk melihat hasil pola untuk bidang tertentu, seperti bidang yang berasal dari `parse` perintah, gunakan`pattern @fieldname`.
- Kueri dengan output non-log, seperti kueri dengan `stats` perintah, tidak mengembalikan hasil pola.

Bandingkan (diff) dengan rentang waktu sebelumnya

Anda dapat menggunakan Wawasan CloudWatch Log untuk membandingkan perubahan dalam peristiwa log Anda dari waktu ke waktu. Anda dapat membandingkan peristiwa log yang dicerna selama rentang waktu terakhir dengan log dari periode waktu sebelumnya. Atau, Anda dapat membandingkan dengan periode waktu sebelumnya yang serupa. Ini dapat membantu Anda menemukan apakah kesalahan dalam log Anda baru-baru ini diperkenalkan atau sudah terjadi, dan dapat membantu Anda menemukan tren lainnya.

Kueri perbandingan hanya mengembalikan pola dalam hasil, bukan peristiwa log mentah. Pola yang dikembalikan akan membantu Anda dengan cepat melihat tren dan perubahan dalam peristiwa log dari waktu ke waktu. Setelah Anda menjalankan kueri perbandingan dan mendapatkan hasil

pola, Anda dapat melihat contoh peristiwa log mentah untuk pola yang Anda minati. Untuk informasi selengkapnya tentang pola log, lihatAnalisis pola.

Saat Anda menjalankan kueri perbandingan, kueri Anda dianalisis terhadap dua periode waktu yang berbeda: periode kueri asli yang Anda pilih, dan periode perbandingan. Periode perbandingan selalu sama panjangnya dengan periode kueri asli Anda. Interval waktu default untuk perbandingan adalah sebagai berikut.

- Periode sebelumnya — Membandingkan dengan periode waktu sebelum periode waktu kueri Anda.
- Hari sebelumnya — Membandingkan dengan periode waktu satu hari sebelum periode waktu kueri Anda.
- Minggu sebelumnya — Bandingkan dengan periode waktu satu minggu sebelum periode waktu kueri Anda.
- Bulan sebelumnya — Membandingkan dengan periode waktu satu bulan sebelum periode waktu kueri Anda.

> **Note**
>
> Kueri yang menggunakan perbandingan dikenakan biaya yang mirip dengan menjalankan satu kueri Wawasan CloudWatch Log selama rentang waktu gabungan. Untuk informasi selengkapnya, lihat CloudWatch Harga Amazon.

Untuk menjalankan kueri perbandingan

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

   Kueri default muncul di kotak kueri.
3. Simpan kueri default atau masukkan kueri yang berbeda.
4. Di menu drop-down Pilih grup log, pilih satu atau beberapa grup log untuk kueri.
5. (Opsional) Gunakan pemilih interval waktu untuk memilih periode waktu yang ingin Anda kueri. Kueri default adalah untuk jam data log sebelumnya.
6. Dengan pemilih rentang waktu, pilih Bandingkan. Kemudian pilih periode waktu sebelumnya yang ingin Anda bandingkan dengan log asli, dan pilih Terapkan.

7. Pilih Run query (Jalankan kueri).

   Untuk menyebabkan kueri mengambil data dari periode perbandingan, `diff` perintah ditambahkan ke kueri Anda.

8. Pilih tab Patterns untuk melihat hasilnya.

   Tabel menampilkan informasi berikut:

   - Setiap Pola, dengan bagian variabel dari pola digantikan oleh simbol token dinamis`<*>`. Untuk informasi selengkapnya, lihat Analisis pola.
   - Jumlah peristiwa adalah jumlah peristiwa log dengan pola itu dalam periode waktu asli yang lebih terkini.
   - Perbedaan jumlah peristiwa adalah perbedaan antara jumlah peristiwa log yang cocok dalam periode waktu saat ini versus periode waktu perbandingan. Berbeda positif berarti ada lebih banyak peristiwa seperti itu dalam periode waktu saat ini.
   - Deskripsi perbedaan secara singkat merangkum perubahan pola itu antara periode waktu saat ini dan periode perbandingan.
   - Jenis keparahan adalah kemungkinan tingkat keparahan peristiwa log dengan pola ini, berdasarkan kata-kata yang ditemukan dalam peristiwa log sepertiFATAL,ERROR, dan. WARN

9. Untuk memeriksa lebih lanjut salah satu pola dalam daftar, pilih ikon di kolom Inspect untuk salah satu pola.

   Panel pemeriksaan Pola muncul dan menampilkan yang berikut:

   - Pola. Pilih token dalam pola untuk menganalisis nilai token tersebut.
   - Histogram yang menunjukkan jumlah kemunculan pola selama rentang waktu yang ditanyakan. Ini dapat membantu Anda mengidentifikasi tren menarik seperti peningkatan tiba-tiba dalam terjadinya suatu pola.
   - Tab Log samples menampilkan beberapa peristiwa log yang cocok dengan pola yang dipilih.
   - Tab Nilai Token menampilkan nilai token dinamis yang dipilih, jika Anda telah memilihnya.

   > **Note**
   >
   > Maksimal 10 nilai token ditangkap untuk setiap token. Jumlah token mungkin tidak tepat. CloudWatch Log menggunakan penghitung probabilistik untuk menghasilkan jumlah token, bukan nilai absolut.

- Tab pola Terkait menampilkan pola lain yang sering terjadi mendekati waktu yang sama dengan pola yang Anda periksa. Misalnya, jika pola untuk `ERROR` pesan biasanya disertai dengan peristiwa log lain yang ditandai `INFO` dengan detail tambahan, pola itu ditampilkan di sini.

Kueri Sampel

Bagian ini berisi daftar perintah kueri umum dan berguna yang dapat Anda jalankan di CloudWatch konsol. Untuk informasi tentang cara menjalankan perintah kueri, lihat Tutorial: Menjalankan dan memodifikasi contoh kueri di Panduan Pengguna Amazon CloudWatch Logs.

Untuk informasi selengkapnya tentang sintaks kueri, lihatCloudWatch Sintaks kueri Log Insights.

Topik

Kueri umum

Temukan 25 peristiwa log yang paling baru ditambahkan.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

Dapatkan daftar jumlah pengecualian per jam.

```
filter @message like /Exception/
```

```
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

Dapatkan daftar peristiwa log yang bukan pengecualian.

```
fields @message | filter @message not like /Exception/
```

Dapatkan peristiwa log terbaru untuk setiap nilai unik **`server`** bidang.

```
fields @timestamp, server, severity, message
| sort @timestamp asc
| dedup server
```

Dapatkan peristiwa log terbaru untuk setiap nilai unik **`server`** bidang untuk setiap **`severity`** jenis.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server, severity
```

Kueri untuk log Lambda

Tentukan jumlah memori yang dilebih-lebihkan.

```
filter @type = "REPORT"
    | stats max(@memorySize / 1000 / 1000) as provisonedMemoryMB,
        min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

Buat laporan latensi.

```
filter @type = "REPORT" |
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

Cari pemanggilan fungsi lambat, dan hilangkan permintaan duplikat yang dapat muncul dari percobaan ulang atau kode sisi klien. Dalam query ini, **`@duration`** adalah dalam milidetik.

```
fields @timestamp, @requestId, @message, @logStream
| filter @type = "REPORT" and @duration > 1000
| sort @timestamp desc
| dedup @requestId
| limit 20
```

Kueri untuk log aliran VPC Amazon

Temukan 15 transfer paket teratas di seluruh host:

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

Temukan transfer 15 byte teratas untuk host pada subnet tertentu.

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

Temukan alamat IP yang menggunakan UDP sebagai protokol transfer data.

```
filter protocol=17 | stats count(*) by srcAddr
```

Temukan alamat IP tempat catatan aliran dilewati selama jendela pengambilan.

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

Temukan satu catatan untuk setiap koneksi, untuk membantu memecahkan masalah konektivitas jaringan.

```
fields @timestamp, srcAddr, dstAddr, srcPort, dstPort, protocol, bytes
| filter logStream = 'vpc-flow-logs' and interfaceId = 'eni-0123456789abcdef0'
```

```
| sort @timestamp desc
| dedup srcAddr, dstAddr, srcPort, dstPort, protocol
| limit 20
```

Kueri untuk log Route 53

Temukan distribusi catatan per jam berdasarkan jenis kueri.

```
stats count(*) by queryType, bin(1h)
```

Temukan 10 DNS resolver dengan jumlah permintaan tertinggi.

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

Temukan jumlah catatan berdasarkan domain dan subdomain di mana server gagal menyelesaikan permintaan DNS.

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

Kueri untuk log CloudTrail

Temukan jumlah entri log untuk setiap layanan, jenis acara, dan AWS Wilayah.

```
stats count(*) by eventSource, eventName, awsRegion
```

Temukan host Amazon EC2 yang dimulai atau dihentikan di Wilayah tertentu AWS .

```
filter (eventName="StartInstances" or eventName="StopInstances") and awsRegion="us-
east-2"
```

Temukan AWS Wilayah, nama pengguna, dan ARN pengguna IAM yang baru dibuat.

```
filter eventName="CreateUser"
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

Temukan jumlah catatan di mana pengecualian terjadi saat menjalankan API**`UpdateTrail`**.

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

Temukan entri log di mana TLS 1.0 atau 1.1 digunakan

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by userIdentity.accountId, recipientAccountId,
 eventSource, eventName, awsRegion, tlsDetails.tlsVersion, tlsDetails.cipherSuite,
 userAgent
| sort eventSource, eventName, awsRegion, tlsDetails.tlsVersion
```

Temukan jumlah panggilan per layanan yang menggunakan TLS versi 1.0 atau 1.1

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by eventSource
| sort numOutdatedTlsCalls desc
```

Pertanyaan untuk Amazon API Gateway

Temukan 10 kesalahan 4XX terakhir

```
fields @timestamp, status, ip, path, httpMethod
| filter status>=400 and status<=499
| sort @timestamp desc
| limit 10
```

Identifikasi 10 Amazon API Gateway permintaan yang paling lama berjalan di grup log akses Anda Amazon API Gateway

```
fields @timestamp, status, ip, path, httpMethod, responseLatency
| sort responseLatency desc
| limit 10
```

Kembalikan daftar jalur API paling populer di grup log Amazon API Gateway akses Anda

```
stats count(*) as requestCount by path
| sort requestCount desc
| limit 10
```

Membuat laporan latensi integrasi untuk grup log Amazon API Gateway akses Anda

```
filter status=200
| stats avg(integrationLatency), max(integrationLatency),
min(integrationLatency) by bin(1m)
```

Pertanyaan untuk gateway NAT

Jika Anda melihat biaya yang lebih tinggi dari biasanya dalam AWS tagihan Anda, Anda dapat menggunakan Wawasan CloudWatch Log untuk menemukan kontributor teratas. Untuk informasi selengkapnya tentang perintah kueri berikut, lihat Bagaimana cara menemukan kontributor teratas untuk lalu lintas melalui gateway NAT di VPC saya? di halaman dukungan AWS premium.

> **Note**
>
> Dalam perintah kueri berikut, ganti “x.x.x.x” dengan IP pribadi gateway NAT Anda, dan ganti “y.y” dengan dua oktet pertama dari rentang CIDR VPC Anda.

Temukan contoh yang mengirimkan lalu lintas terbanyak melalui gateway NAT Anda.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Tentukan lalu lintas yang menuju dan dari instance di gateway NAT Anda.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.') or (srcAddr like 'xxx.xx.xx.xx'
 and dstAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Tentukan tujuan internet yang paling sering berkomunikasi dengan instance di VPC Anda untuk upload dan download.

Untuk upload

```
filter (srcAddr like 'x.x.x.x' and dstAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Untuk unduhan

```
filter (dstAddr like 'x.x.x.x' and srcAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Kueri untuk log server Apache

Anda dapat menggunakan Wawasan CloudWatch Log untuk menanyakan log server Apache. Untuk informasi selengkapnya tentang kueri berikut, lihat Menyederhanakan log server Apache dengan Wawasan CloudWatch Log di Blog Operasi & Migrasi AWS Cloud.

Temukan bidang yang paling relevan, sehingga Anda dapat meninjau log akses Anda dan memeriksa lalu lintas di jalur /admin aplikasi Anda.

```
fields @timestamp, remoteIP, request, status, filename| sort @timestamp desc
| filter filename="/var/www/html/admin"
| limit 20
```

Temukan nomor permintaan GET unik yang mengakses halaman utama Anda dengan kode status “200" (sukses).

```
fields @timestamp, remoteIP, method, status
| filter status="200" and referrer= http://34.250.27.141/ and method= "GET"
| stats count_distinct(remoteIP) as UniqueVisits
| limit 10
```

Temukan berapa kali layanan Apache Anda dimulai ulang.

```
fields @timestamp, function, process, message
| filter message like "resuming normal operations"
| sort @timestamp desc
| limit 20
```

Kueri untuk Amazon EventBridge

Dapatkan jumlah EventBridge acara yang dikelompokkan berdasarkan jenis detail acara

```
fields @timestamp, @message
| stats count(*) as numberOfEvents by `detail-type`
| sort numberOfEvents desc
```

Contoh perintah parse

Gunakan ekspresi glob untuk mengekstrak bidang**@user**,**@method**, dan **@latency** dari bidang log **@message** dan kembalikan latensi rata-rata untuk setiap kombinasi unik dan. **@method** **@user**

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Gunakan ekspresi reguler untuk mengekstrak bidang**@user2**,**@method2**, dan **@latency2** dari bidang log **@message** dan kembalikan latensi rata-rata untuk setiap kombinasi unik **@method2** dan**@user2**.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Mengekstrak bidang**loggingTime**, **loggingType** dan**loggingMessage**, memfilter ke log peristiwa yang berisi **ERROR** atau **INFO** string, dan kemudian hanya menampilkan **loggingMessage** dan **loggingType** bidang untuk peristiwa yang berisi **ERROR** string.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

Visualisasikan data log dalam grafik

Anda dapat menggunakan visualisasi seperti diagram batang, diagram garis, dan bagan area bertumpuk untuk mengidentifikasi pola dalam data log Anda dengan lebih efisien. CloudWatch Log Insights menghasilkan visualisasi untuk kueri yang menggunakan `stats` fungsi dan satu atau beberapa fungsi agregasi. Untuk informasi lebih lanjut, lihat statistik.

Simpan dan jalankan kembali kueri CloudWatch Logs Insights

Setelah Anda membuat kueri, Anda dapat menyimpannya, dan menjalankannya lagi nanti. Kueri disimpan dalam struktur folder, sehingga Anda dapat mengaturnya. Anda dapat menyimpan sebanyak 1000 kueri per wilayah dan per akun.

Untuk menyimpan kueri, Anda harus masuk ke peran yang memiliki izin`logs:PutQueryDefinition`. Untuk melihat daftar kueri yang disimpan, Anda harus masuk ke peran yang memiliki izin`logs:DescribeQueryDefinitions`.

Untuk menyimpan kueri

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Buat kueri di editor kueri.
4. Pilih Simpan.

   Jika Anda tidak melihat tombol Simpan, Anda perlu mengubah ke desain baru untuk konsol CloudWatch Log. Untuk melakukannya:

   a. Pada panel navigasi, pilih Grup log.
   b. Pilih Try the new design (Coba desain baru).
   c. Di panel navigasi, pilih Insights (Wawasan) dan kembali ke langkah 3 dalam prosedur ini.
5. Masukkan nama untuk kueri.
6. (Opsional) Pilih folder tempat Anda ingin menyimpan kueri. Pilih Create new (Buat baru) untuk membuat folder. Jika Anda membuat folder baru, Anda dapat menggunakan karakter garis miring (/) dalam nama folder untuk menentukan struktur folder. Sebagai contoh, menamai folder baru dengan **`folder-level-1/folder-level-2`** akan membuat folder tingkat atas yang disebut **`folder-level-1`**, dengan folder lain yang bernama **`folder-level-2`** di dalam folder itu. Kueri disimpan dalam **`folder-level-2`**.

7. (Opsional) Ubah grup log kueri atau teks kueri.
8. Pilih Simpan.

> **Tip**
>
> Anda dapat membuat folder untuk kueri yang disimpan dengan`PutQueryDefinition`. Untuk membuat folder untuk kueri yang disimpan, gunakan garis miring (/) untuk mengawali nama kueri yang Anda inginkan dengan nama folder yang Anda inginkan:. <*`folder-name`*>/<*`query-name`*> Untuk informasi lebih lanjut tentang tindakan ini, lihat PutQueryDefinition.

Untuk menjalankan kueri yang disimpan

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Di sebelah kanan, pilih Queries (Kueri).
4. Pilih kueri dari daftar Saved queries (Kueri tersimpan). Itu akan muncul di editor kueri.
5. Pilih Jalankan.

Untuk menyimpan versi baru dari kueri tersimpan

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Di sebelah kanan, pilih Queries (Kueri).
4. Pilih kueri dari daftar Saved queries (Kueri tersimpan). Itu akan muncul di editor kueri.
5. Modifikasi kueri. Jika Anda perlu menjalankannya untuk memeriksa pekerjaan Anda, pilih Run query (Jalankan kueri).
6. Saat Anda siap untuk menyimpan versi baru, pilih Actions (Tindakan), Save as (Simpan sebagai).
7. Masukkan nama untuk kueri.
8. (Opsional) Pilih folder tempat Anda ingin menyimpan kueri. Pilih Create new (Buat baru) untuk membuat folder. Jika Anda membuat folder baru, Anda dapat menggunakan karakter garis miring (/) dalam nama folder untuk menentukan struktur folder. Sebagai contoh, menamai folder baru dengan **`folder-level-1/folder-level-2`** akan membuat folder tingkat atas yang disebut

`folder-level-1`, dengan folder lain yang bernama **`folder-level-2`** di dalam folder itu. Kueri disimpan dalam **`folder-level-2`**.

9. (Opsional) Ubah grup log kueri atau teks kueri.
10. Pilih Simpan.

Untuk menghapus kueri, Anda harus masuk ke peran yang memiliki izin `logs:DeleteQueryDefinition`.

Untuk mengedit atau menghapus kueri tersimpan

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Di sebelah kanan, pilih Queries (Kueri).
4. Pilih kueri dari daftar Saved queries (Kueri tersimpan). Itu akan muncul di editor kueri.
5. Pilih Actions (Tindakan), Edit atau Actions (Tindakan), Delete (Hapus).

Tambahkan kueri ke dasbor atau ekspor hasil kueri

Setelah menjalankan kueri, Anda dapat menambahkan kueri ke CloudWatch dasbor atau menyalin hasilnya ke clipboard.

Kueri yang ditambahkan ke dasbor akan dijalankan setiap kali Anda memuat dasbor dan setiap kali dasbor disegarkan. Kueri ini dihitung terhadap batas 30 kueri Wawasan CloudWatch Log bersamaan.

Untuk menambahkan hasil kueri ke dasbor

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Pilih satu atau beberapa grup log dan jalankan kueri.
4. Pilih Tambahkan ke dasbor.
5. Pilih dasbor, atau pilih Create new (Buat baru) untuk membuat dasbor untuk hasil kueri.
6. Pilih jenis widget yang akan digunakan untuk hasil kueri.
7. Masukkan nama untuk widget.
8. Pilih Tambahkan ke dasbor.

Untuk menyalin hasil kueri ke clipboard atau mengunduh hasil kueri

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Pilih satu atau beberapa grup log dan jalankan kueri.
4. Pilih Export results (Ekspor hasil), lalu pilih opsi yang Anda inginkan.

Lihat kueri atau riwayat kueri yang sedang berjalan

Anda dapat melihat kueri yang sedang berlangsung serta riwayat kueri terbaru Anda.

Kueri yang sedang berjalan mencakup kueri yang telah ditambahkan ke dasbor. Anda dibatasi hingga 30 kueri Wawasan CloudWatch Log bersamaan per akun, termasuk kueri yang ditambahkan ke dasbor.

Untuk melihat riwayat kueri terbaru Anda

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, lalu pilih Wawasan Log.
3. Pilih Riwayat, jika Anda menggunakan desain baru untuk konsol CloudWatch Log. Jika Anda menggunakan desain lama, pilih Actions (Tindakan), View query history for this account (Lihat riwayat kueri untuk akun ini).

   Daftar kueri terbaru Anda akan muncul. Anda dapat menjalankan kembali salah satunya dengan memilih kueri dan memilih Run (Jalankan).

   Di bawah Status, CloudWatch Log ditampilkan Sedang berlangsung untuk kueri apa pun yang sedang berjalan.

Enkripsi hasil kueri dengan AWS Key Management Service

Secara default, CloudWatch Log mengenkripsi hasil tersimpan dari kueri Wawasan CloudWatch Log Anda menggunakan metode enkripsi sisi server CloudWatch Log default. Anda dapat memilih untuk menggunakan AWS KMS kunci untuk mengenkripsi hasil ini sebagai gantinya. Jika Anda mengaitkan AWS KMS kunci dengan hasil enkripsi Anda, maka CloudWatch Log menggunakan kunci tersebut untuk mengenkripsi hasil yang disimpan dari semua kueri di akun.

Jika nanti Anda memisahkan kunci dari hasil kueri, CloudWatch Log akan kembali ke metode enkripsi default untuk kueri selanjutnya. Tetapi kueri yang berjalan saat kunci dikaitkan masih dienkripsi dengan kunci itu. CloudWatch Log masih dapat mengembalikan hasil tersebut setelah kunci KMS dipisahkan, karena CloudWatch Log masih dapat terus mereferensikan kunci. Namun, jika kunci kemudian dinonaktifkan, maka CloudWatch Log tidak dapat membaca hasil kueri yang dienkripsi dengan kunci itu.

> **Important**
>
> CloudWatch Log hanya mendukung kunci KMS simetris. Jangan gunakan kunci asimetris untuk mengenkripsi hasil kueri Anda. Untuk informasi selengkapnya, lihat Menggunakan Kunci Simetris dan Asimetris.

Batas

- Untuk melakukan langkah-langkah berikut, Anda harus memiliki izin berikut: `kms:CreateKey`, `kms:GetKeyPolicy`, dan `kms:PutKeyPolicy`.
- Setelah Anda mengaitkan atau memisahkan kunci dari hasil kueri Anda, diperlukan waktu hingga lima menit agar operasi diterapkan.
- Jika Anda mencabut akses CloudWatch Log ke kunci terkait atau menghapus kunci KMS terkait, data terenkripsi Anda di CloudWatch Log tidak dapat diambil lagi.
- Anda tidak dapat menggunakan CloudWatch konsol untuk mengaitkan kunci, Anda harus menggunakan AWS CLI atau CloudWatch Logs API.

Langkah 1: Buat AWS KMS key

Untuk membuat kunci KMS gunakan perintah create-key berikut:

```
aws kms create-key
```

Output berisi ID kunci dan Amazon Resource Name (ARN) dari kunci. Berikut ini adalah output contoh:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
```

```
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Langkah 2: Tetapkan izin pada tombol KMS

Secara default, semua kunci KMS bersifat pribadi. Hanya pemilik sumber daya yang dapat menggunakannya untuk mengenkripsi dan mendekripsi data. Namun, pemilik sumber daya dapat memberikan izin untuk mengakses kunci ke pengguna dan sumber daya lain. Dengan langkah ini, Anda memberikan izin utama layanan CloudWatch Log untuk menggunakan kunci. Prinsipal layanan ini harus berada di AWS Wilayah yang sama di mana kunci disimpan.

Sebagai praktik terbaik, kami menyarankan Anda membatasi penggunaan kunci hanya untuk AWS akun yang Anda tentukan.

Pertama, simpan kebijakan default untuk kunci KMS Anda seperti `policy.json` menggunakan get-key-policyperintah berikut:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Buka file `policy.json` di editor teks dan tambahkan bagian dalam huruf tebal dari salah satu pernyataan berikut. Pisahkan pernyataan yang ada dari pernyataan baru dengan koma. Pernyataan ini menggunakan `Condition` bagian untuk meningkatkan keamanan AWS KMS kunci. Untuk informasi selengkapnya, lihat AWS KMS kunci dan konteks enkripsi.

`Condition`Bagian dalam contoh ini membatasi penggunaan AWS KMS kunci untuk hasil kueri Wawasan CloudWatch Log di akun yang ditentukan.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:logs:region:account_ID:query-result:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "Your_account_ID"
                }
            }
        }
    ]
}
```

Terakhir, tambahkan kebijakan yang diperbarui menggunakan put-key-policyperintah berikut:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Langkah 3: Kaitkan kunci KMS dengan hasil kueri Anda

Untuk mengaitkan kunci KMS dengan hasil kueri di akun

Gunakan disassociate-kms-keyperintah sebagai berikut:

```
aws logs associate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-result:*" --kms-key-id "key-arn"
```

Langkah 4: Lepaskan kunci dari hasil kueri di akun

Untuk memisahkan kunci KMS yang terkait dengan hasil kueri, gunakan perintah berikut: disassociate-kms-key

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-result:*"
```

Gunakan bahasa alami untuk menghasilkan dan memperbarui kueri Wawasan CloudWatch Log

> **Note**
>
> Fitur ini umumnya tersedia di AS Timur (Virginia N.), AS Barat (Oregon), dan Asia Pasifik (Tokyo) untuk CloudWatch Log.

CloudWatch Log mendukung kemampuan kueri bahasa alami untuk membantu Anda menghasilkan dan memperbarui kueri untuk Wawasan CloudWatch Log dan Wawasan CloudWatch Metrik.

Dengan kemampuan ini, Anda dapat mengajukan pertanyaan tentang atau menjelaskan data CloudWatch Log yang Anda cari dalam bahasa Inggris biasa. Kemampuan bahasa alami menghasilkan kueri berdasarkan prompt yang Anda masukkan dan memberikan line-by-line penjelasan tentang cara kerja kueri. Anda juga dapat memperbarui kueri Anda untuk menyelidiki lebih lanjut data Anda.

Bergantung pada lingkungan Anda, Anda dapat memasukkan petunjuk seperti “Apa 100 alamat IP sumber teratas berdasarkan byte yang ditransfer?” dan “Temukan 10 permintaan fungsi Lambda paling lambat.”

Untuk menghasilkan kueri Wawasan CloudWatch Log dengan kemampuan ini, buka editor kueri Wawasan CloudWatch Log, pilih grup log yang ingin Anda kueri, dan pilih Buat kueri.

> **Important**
>
> Untuk menggunakan kemampuan kueri bahasa alami, Anda harus menggunakan CloudWatchLogsFullAccess, CloudWatchLogsReadOnlyAccess, AdministratorAccess, atau ReadOnlyAccesskebijakan.
> Anda juga dapat menyertakan tindakan `cloudwatch:GenerateQuery` dalam kebijakan terkelola atau inline pelanggan baru atau yang sudah ada.

Kueri contoh

Contoh di bagian ini menjelaskan cara menghasilkan dan memperbarui kueri menggunakan kemampuan bahasa alami.

> **Note**
>
> Untuk informasi selengkapnya tentang editor dan sintaks kueri Wawasan CloudWatch Log, lihat sintaks kueri Wawasan CloudWatch Log.

Contoh: Menghasilkan kueri bahasa alami

Untuk menghasilkan kueri menggunakan bahasa alami, masukkan prompt dan pilih Hasilkan kueri baru. Contoh ini menunjukkan kueri yang melakukan pencarian dasar.

Prompt

Berikut ini adalah contoh prompt yang mengarahkan kemampuan untuk mencari 10 pemanggilan fungsi Lambda paling lambat.

```
Find the 10 slowest requests
```

Kueri

Berikut ini contoh kueri yang dihasilkan kemampuan bahasa alami berdasarkan prompt. Perhatikan bagaimana prompt muncul di komentar sebelum kueri. Setelah kueri, Anda dapat membaca penjelasan yang menggambarkan cara kerja kueri.

```
# Find the 10 slowest requests
fields @timestamp, @message, @duration
| sort @duration desc
| limit 10
# This query retrieves the timestamp, message and duration fields from the logs and
 sorts them in descending order by duration to find the 10 slowest requests.
```

> **Note**
>
> Untuk mematikan tampilan prompt Anda dan penjelasan tentang cara kerja kueri, gunakan ikon roda gigi di editor Anda.

Contoh: Memperbarui kueri bahasa alami

Anda dapat memperbarui kueri dengan mengedit prompt awal dan kemudian memilih Perbarui kueri.

Prompt yang diperbarui

Contoh berikut menunjukkan versi yang diperbarui dari prompt sebelumnya. Alih-alih prompt yang mencari 10 pemanggilan fungsi Lambda paling lambat, prompt ini sekarang mengarahkan kemampuan untuk mencari 20 pemanggilan fungsi Lambda paling lambat dan menyertakan kolom lain untuk peristiwa log tambahan.

```
Show top 20 slowest requests instead and display requestId as a column
```

Kueri yang diperbarui

Berikut ini contoh kueri yang diperbarui. Perhatikan bagaimana prompt muncul di komentar sebelum kueri yang diperbarui. Setelah kueri, Anda dapat membaca penjelasan yang menggambarkan bagaimana kueri asli diperbarui.

```
# Show top 20 slowest requests instead and display requestId as a column
fields @timestamp, @message, @requestId, @duration
| sort @duration desc
| limit 20
# This query modifies the original query by replacing the @message field with the
 @requestId field and changing the limit from 10 to 20 to return the top 20 log events
 by duration instead of the top 10.
```

Memilih untuk tidak menggunakan data Anda untuk perbaikan layanan

Data prompt bahasa alami yang Anda berikan untuk melatih model AI dan menghasilkan kueri yang relevan digunakan semata-mata untuk menyediakan dan memelihara layanan Anda. Data ini dapat digunakan untuk meningkatkan kualitas Wawasan CloudWatch Log. Kepercayaan dan privasi Anda, serta keamanan konten Anda, menjadi prioritas utama kami. Untuk informasi selengkapnya, silakan lihat AWS Ketentuan Layanan dan AWS kebijakan AI yang bertanggung jawab.

Anda dapat memilih untuk tidak menggunakan konten Anda untuk mengembangkan atau memperbaiki mutu kueri bahasa alami dengan membuat kebijakan penolakan layanan AI. Untuk memilih keluar dari pengumpulan data untuk semua fitur AI CloudWatch Log, termasuk kemampuan pembuatan kueri, Anda harus membuat kebijakan opt-out untuk Log. CloudWatch Untuk informasi selengkapnya, silakan lihat kebijakan penolakan layanan AI di AWS Organizations Panduan Pengguna.

Deteksi anomali log

Anda dapat membuat detektor anomali log untuk setiap grup log. Detektor anomali memindai peristiwa log yang dicerna ke dalam grup log dan menemukan anomali dalam data log. Deteksi anomali menggunakan pembelajaran mesin dan pengenalan pola untuk menetapkan dasar konten log yang khas.

Setelah Anda membuat detektor anomali untuk grup log, ia berlatih menggunakan dua minggu terakhir peristiwa log di grup log untuk pelatihan. Periode pelatihan bisa memakan waktu hingga 15 menit. Setelah pelatihan selesai, ia mulai menganalisis log masuk untuk mengidentifikasi anomali, dan anomali ditampilkan di konsol CloudWatch Log untuk Anda periksa.

CloudWatch Pengenalan pola log mengekstrak pola log dengan mengidentifikasi konten statis dan dinamis di log Anda. Pola berguna untuk menganalisis kumpulan log besar karena sejumlah besar peristiwa log sering dapat dikompresi menjadi beberapa pola.

Misalnya, lihat contoh berikut dari tiga peristiwa log.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

Dalam sampel sebelumnya, ketiga peristiwa log mengikuti satu pola:

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

Bidang dalam pola disebut token. Bidang yang bervariasi dalam suatu pola, seperti ID permintaan atau stempel waktu, disebut sebagai token dinamis. Token dinamis diwakili oleh <*> saat CloudWatch Log menampilkan pola. Setiap nilai berbeda yang ditemukan untuk token dinamis disebut nilai token.

Contoh umum token dinamis termasuk kode kesalahan, stempel waktu, dan ID permintaan.

Deteksi anomali log menggunakan pola-pola ini untuk menemukan anomali. Setelah periode pelatihan model detektor anomali, log dievaluasi terhadap tren yang diketahui. Detektor anomali menandai fluktuasi yang signifikan sebagai anomali.

Membuat detektor anomali log tidak menimbulkan biaya.

Tingkat keparahan dan prioritas anomali dan pola

Setiap anomali yang ditemukan oleh detektor anomali log diberi prioritas. Setiap pola yang ditemukan diberi tingkat keparahan.

- Prioritas dihitung secara otomatis, dan didasarkan pada tingkat keparahan pola dan jumlah penyimpangan dari nilai yang diharapkan. Misalnya, jika nilai token tertentu tiba-tiba meningkat sebesar 500%, anomali itu mungkin ditetapkan sebagai `HIGH` prioritas bahkan jika tingkat keparahannya. `NONE`
- Tingkat keparahan hanya didasarkan pada kata kunci yang ditemukan dalam pola seperti`FATAL`,`ERROR`, dan`WARN`. Jika tidak satu pun dari kata kunci ini ditemukan, tingkat keparahan pola ditandai sebagai`NONE`.

Waktu visibilitas anomali

Saat Anda membuat detektor anomali, Anda menentukan periode visibilitas anomali maksimum untuknya. Ini adalah jumlah hari anomali ditampilkan di konsol dan dikembalikan oleh operasi ListAnomaliesAPI. Setelah periode waktu ini berlalu untuk anomali, jika terus terjadi, itu secara otomatis diterima sebagai perilaku biasa dan model detektor anomali berhenti menandainya sebagai anomali.

Jika Anda tidak menyesuaikan waktu visibilitas saat membuat detektor anomali, 21 hari digunakan sebagai default.

Menekan anomali

Setelah anomali ditemukan, Anda dapat memilih untuk menekannya sementara atau permanen. Menekan anomali menyebabkan detektor anomali berhenti menandai kejadian ini sebagai anomali untuk jumlah waktu yang Anda tentukan. Ketika Anda menekan anomali, Anda dapat memilih untuk menekan hanya anomali spesifik itu, atau menekan semua anomali yang terkait dengan pola di mana anomali itu ditemukan.

Anda masih dapat melihat anomali yang ditekan di konsol. Anda juga dapat memilih untuk berhenti menekannya.

Pertanyaan umum

Apakah AWS menggunakan data saya untuk melatih algoritme pembelajaran mesin untuk AWS digunakan atau untuk pelanggan lain?

Tidak. Model deteksi anomali yang dibuat oleh pelatihan didasarkan pada peristiwa log dalam grup log dan hanya digunakan dalam grup log itu dan akun itu AWS .

Jenis peristiwa log apa yang bekerja dengan baik dengan deteksi anomali?

Deteksi anomali log sangat cocok untuk: Log aplikasi dan jenis log lainnya di mana sebagian besar entri log sesuai dengan pola tipikal. Grup log dengan peristiwa yang berisi tingkat log atau kata kunci tingkat keparahan seperti INFO, ERROR, dan DEBUG sangat cocok untuk mencatat deteksi anomali.

Deteksi anomali log tidak cocok untuk: Peristiwa log dengan struktur JSON yang sangat panjang, seperti Log. CloudTrail Analisis pola hanya menganalisis hingga 1500 karakter pertama dari garis log, sehingga karakter apa pun di luar batas itu dilewati.

Audit atau log akses, seperti log aliran VPC, juga akan kurang berhasil dengan deteksi anomali. Deteksi anomali dimaksudkan untuk menemukan masalah aplikasi, jadi mungkin tidak cocok untuk anomali jaringan atau akses.

Untuk membantu Anda menentukan apakah detektor anomali cocok untuk grup log tertentu, gunakan analisis pola CloudWatch Log untuk menemukan jumlah pola dalam peristiwa log dalam grup. Jika jumlah pola tidak lebih dari sekitar 300, deteksi anomali mungkin bekerja dengan baik. Untuk informasi lebih lanjut tentang analisis pola, lihatAnalisis pola.

Apa yang ditandai sebagai anomali?

Kejadian berikut dapat menyebabkan peristiwa log ditandai sebagai anomali:

- Peristiwa log dengan pola yang tidak terlihat sebelumnya di grup log.
- Variasi yang signifikan terhadap pola yang diketahui.
- Nilai baru untuk token dinamis yang memiliki serangkaian nilai biasa yang terpisah.
- Perubahan besar dalam jumlah kemunculan nilai untuk token dinamis.

Meskipun semua item sebelumnya mungkin ditandai sebagai anomali, itu tidak semua berarti bahwa aplikasi berkinerja buruk. Misalnya, higher-than-usual sejumlah nilai `200` keberhasilan

mungkin ditandai sebagai anomali. Dalam kasus seperti ini, Anda mungkin mempertimbangkan untuk menekan anomali ini yang tidak menunjukkan masalah.

Apa yang terjadi dengan data sensitif yang sedang disembunyikan?

Setiap bagian dari peristiwa log yang disamarkan sebagai data sensitif tidak dipindai untuk anomali. Untuk informasi selengkapnya tentang menyembunyikan data sensitif, lihat Membantu melindungi data log sensitif dengan masking.

Aktifkan deteksi anomali pada grup log

Gunakan langkah-langkah berikut untuk menggunakan CloudWatch konsol untuk membuat detektor anomali log yang memindai grup log untuk mengetahui anomali.

Anda juga dapat membuat detektor anomali secara terprogram. Untuk informasi lebih lanjut, lihat CreateLogAnomalyDetector.

Untuk membuat detektor anomali log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pilih Log, Log Anomali.
3. Pilih Buat detektor anomali.
4. Pilih grup log untuk membuat detektor anomali ini.
5. Masukkan nama untuk detektor dalam nama detektor anomali.
6. (Opsional) Ubah frekuensi Evaluasi dari default 5 menit. Tetapkan nilai ini sesuai dengan frekuensi grup log menerima log baru. Misalnya, jika grup log menerima peristiwa log baru dalam batch setiap 10 menit, maka pengaturan frekuensi evaluasi menjadi 15 menit mungkin tepat.
7. (Opsional) Untuk mengonfigurasi detektor anomali untuk mencari anomali hanya dalam peristiwa log yang berisi kata atau string tertentu, pilih Pola filter.

   Kemudian, masukkan pola dalam pola filter deteksi anomali. Untuk informasi lebih lanjut tentang sintaks pola,Filter sintaks pola untuk filter metrik, filter langganan, peristiwa log filter, dan Live Tail.

   (Opsional) Untuk menguji pola filter Anda, masukkan beberapa pesan log ke dalam pesan peristiwa Log dan kemudian pilih Pola Uji.
8. (Opsional) Untuk mengubah periode visibilitas anomali dari default atau untuk mengaitkan AWS KMS kunci dengan detektor anomali ini, pilih Konfigurasi lanjutan.

a. Untuk mengubah periode visibilitas anomali dari default, masukkan nilai baru di Periode visibilitas anomali maksimum (hari).

b. Untuk mengaitkan AWS KMS kunci dengan detektor anomali ini, masukkan ARN di ARN kunci KMS. Jika Anda menetapkan kunci, informasi anomali yang ditemukan oleh detektor ini dienkripsi saat istirahat dengan kunci. Pengguna harus memiliki izin untuk kunci ini dan detektor anomali untuk mengambil informasi tentang anomali yang ditemukannya.

   Anda juga harus memastikan bahwa kepala layanan CloudWatch Log memiliki izin untuk menggunakan kunci. Untuk informasi selengkapnya, lihat Enkripsi detektor anomali dan hasilnya dengan AWS KMS.

9. Pilih Aktifkan Deteksi Anomali.

   Detektor anomali dibuat dan mulai melatih modelnya, berdasarkan peristiwa log yang dicerna grup log. Setelah sekitar 15 menit, deteksi anomali aktif dan mulai menemukan dan anomali permukaan.

Lihat anomali yang telah ditemukan

Setelah Anda membuat satu atau lebih detektor anomali log, Anda dapat menggunakan CloudWatch konsol untuk melihat anomali yang mereka temukan.

Anda dapat melihat anomali secara terprogram. Untuk informasi lebih lanjut, lihat ListAnomalies.

Untuk melihat anomali yang ditemukan oleh semua detektor anomali log Anda

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pilih Log, Log Anomali.

   Tabel anomali Log muncul. Nomor di bagian atas di sebelah anomali Log menampilkan berapa banyak anomali log yang tercantum dalam tabel. Setiap baris dalam tabel menampilkan informasi berikut:

   - Kolom Anomali menampilkan ringkasan singkat anomali. Ringkasan ini dihasilkan oleh CloudWatch Log.
   - Prioritas anomali. Prioritas dihitung secara otomatis berdasarkan jumlah perubahan dalam peristiwa log, kata kunci seperti yang `Exception` terjadi dalam peristiwa log, dan banyak lagi.

- Pola Log yang menjadi dasar anomali. Untuk informasi lebih lanjut tentang pola, lihatDeteksi anomali log.
- Tren log anomali menampilkan histogram yang menggambarkan volume log yang cocok dengan pola.
- Waktu deteksi terakhir menampilkan waktu terbaru anomali ini ditemukan.
- Waktu deteksi pertama menunjukkan pertama kali anomali ini ditemukan.
- Detektor anomali menampilkan nama grup log yang berisi peristiwa log yang terkait dengan anomali ini. Anda dapat memilih nama ini untuk melihat halaman detail grup log.

3. Untuk memeriksa lebih lanjut satu anomali, pilih tombol radio di barisnya.

   Panel pemeriksaan Pola muncul dan menampilkan yang berikut:

   - Pola yang menjadi dasar anomali ini. Pilih token dalam pola untuk menganalisis nilai token tersebut.
   - Histogram yang menunjukkan jumlah kemunculan anomali selama rentang waktu yang ditanyakan.
   - Tab sampel Log menampilkan beberapa peristiwa log yang merupakan bagian dari anomali.
   - Tab Nilai Token menampilkan nilai token dinamis yang dipilih, jika Anda telah memilihnya.

   > **Note**
   >
   > Maksimal 10 nilai token ditangkap untuk setiap token. Jumlah token mungkin tidak tepat. CloudWatch Log menggunakan penghitung probabilistik untuk menghasilkan jumlah token, bukan nilai absolut.

4. Untuk menekan anomali, pilih tombol radio di barisnya lalu lakukan hal berikut:

   a. Pilih Tindakan, Menekan Anomali.

   b. Kemudian tentukan berapa lama Anda ingin anomali ditekan.

   c. Untuk menekan semua anomali yang terkait dengan pola ini, pilih Suppress Pattern.

   d. Pilih Menekan anomali.

Untuk melihat anomali yang ditemukan dalam satu grup log

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

2. Pilih Log, Grup log.
3. Pilih nama grup log, lalu pilih tab Deteksi anomali.

   Tabel deteksi anomali muncul. Nomor di bagian atas di sebelah anomali Log menampilkan berapa banyak anomali log yang tercantum dalam tabel. Setiap baris dalam tabel menampilkan informasi berikut:

   - Kolom Anomali menampilkan ringkasan singkat anomali. Ringkasan ini dihasilkan oleh CloudWatch Log.
   - Prioritas anomali. Prioritas dihitung secara otomatis berdasarkan jumlah perubahan dalam peristiwa log, kata kunci seperti yang `Exception` terjadi dalam peristiwa log, dan banyak lagi.
   - Pola Log yang menjadi dasar anomali. Untuk informasi lebih lanjut tentang pola, lihatDeteksi anomali log.
   - Tren log anomali menampilkan histogram yang menggambarkan volume log yang cocok dengan pola.
   - Waktu deteksi terakhir menampilkan waktu terbaru anomali ini ditemukan.
   - Waktu deteksi pertama menunjukkan pertama kali anomali ini ditemukan.
4. Untuk memeriksa lebih lanjut satu anomali, pilih tombol radio di barisnya.

   Panel pemeriksaan Pola muncul dan menampilkan yang berikut:

   - Pola yang menjadi dasar anomali ini. Pilih token dalam pola untuk menganalisis nilai token tersebut.
   - Histogram yang menunjukkan jumlah kemunculan anomali selama rentang waktu yang ditanyakan.
   - Tab sampel Log menampilkan beberapa peristiwa log yang merupakan bagian dari anomali.
   - Tab Nilai Token menampilkan nilai token dinamis yang dipilih, jika Anda telah memilihnya.

   > **Note**
   >
   > Maksimal 10 nilai token ditangkap untuk setiap token. Jumlah token mungkin tidak tepat. CloudWatch Log menggunakan penghitung probabilistik untuk menghasilkan jumlah token, bukan nilai absolut.

5. Untuk menekan anomali, pilih tombol radio di barisnya lalu lakukan hal berikut:

   a. Pilih Tindakan, Menekan Anomali.

b. Kemudian tentukan berapa lama Anda ingin anomali ditekan.

c. Untuk menekan semua anomali yang terkait dengan pola ini, pilih Suppress Pattern.

d. Pilih Menekan anomali.

Buat alarm pada detektor anomali log

Anda dapat membuat alarm untuk detektor anomali log di grup log. Anda dapat menentukan agar alarm masuk ke ALARM keadaan ketika sejumlah anomali tertentu ditemukan di grup log selama periode waktu tertentu. Anda juga dapat menggunakan filter sehingga hanya anomali prioritas tertentu yang dihitung oleh alarm.

Untuk membuat alarm untuk detektor anomali log

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

   Tabel detektor anomali log muncul.
3. Pilih tombol radio untuk detektor anomali yang ingin Anda atur alarmnya, dan pilih Buat alarm.

   Wizard pembuatan CloudWatch alarm muncul. LogAnomalyDetectorBidang menampilkan nama detektor anomali yang Anda pilih. Bidang nama metrik ditampilkan AnomalyCount.
4. (Opsional) Untuk memfilter alarm ini untuk prioritas anomali, lakukan salah satu hal berikut:
   - Agar alarm hanya menghitung anomali prioritas tinggi, masukkan untuk. **HIGH** LogAnomalyPriority
   - Agar alarm hanya menghitung anomali prioritas tinggi dan menengah, masukkan. **MEDIUM** LogAnomalyPriority

   Untuk informasi selengkapnya tentang tingkat prioritas, lihatTingkat keparahan dan prioritas anomali dan pola.
5. Pilih untuk menggunakan ambang deteksi anomali statis atau metrik untuk alarm. Pilihan ini menentukan bagaimana ambang alarm diatur. Ambang batas statis berarti bahwa ambang alarm adalah angka statis dan konstan yang Anda pilih. Ambang deteksi anomali berarti CloudWatch menentukan rentang nilai yang biasa, dan alarm memicu jika hitungan aktual melewati ambang batas pita ini. Anda tidak harus memilih Deteksi anomali untuk alarm deteksi anomali log. Untuk informasi lebih lanjut tentang deteksi anomali metrik, lihat Menggunakan deteksi CloudWatch anomali.

6. Untuk Kapanpun ***your-metric-name***. , pilih Lebih Besar, Lebih Hebat/Sama, Lebih Rendah/Sama, atau Lebih Rendah. Kemudian untuk dari . . ., masukkan angka untuk nilai ambang batas Anda. Alarm masuk ke **ALARM** keadaan jika detektor anomali menemukan lebih dari jumlah alarm ini selama waktu yang ditentukan oleh Periode.
7. Pilih Konfigurasi tambahan. Untuk Titik data alarm, tentukan berapa banyak periode evaluasi (titik data) yang harus ada dalam status ALARM untuk memicu alarm. Jika kedua nilai di sini cocok, Anda membuat alarm yang beralih ke status ALARM jika terjadi pelanggaran selama sebanyak itu dalam periode berturut-turut.

   Untuk membuat sebuah alarm M dari N, Anda harus menentukan angka untuk nilai pertama dengan nilai yang lebih rendah dari angka untuk nilai kedua. Untuk informasi selengkapnya, lihat Mengevaluasi alarm.
8. Untuk Perlakuan data yang hilang, pilih perilaku alarm ketika ada beberapa titik data yang hilang. Untuk informasi selengkapnya, lihat Mengonfigurasi cara CloudWatch alarm menangani data yang hilang.
9. Pilih Selanjutnya.
10. Untuk Pemberitahuan, pilih Tambahkan pemberitahuan, lalu tentukan topik Amazon SNS yang akan diberi tahu saat alarm Anda bertransisi keALARM,, OK atau status. INSUFFICIENT_DATA

    a. (Opsional) Untuk mengirimkan beberapa notifikasi untuk status alarm yang sama atau status alarm yang berbeda, silakan pilih Tambahkan notifikasi.

    > **Note**
    >
    > Kami menyarankan Anda untuk menyetel alarm untuk mengambil tindakan ketika alarm beralih statusnya menjadi data tidak mencukupi selain ketika beralih status menjadi Alarm. Hal ini dilakukan karena banyak masalah dengan fungsi Lambda yang terhubung ke sumber data yang dapat menyebabkan alarm beralih statusnya menjadi Data tidak mencukupi.

    b. (Opsional) Jika tidak ingin mengirimkan notifikasi Amazon SNS, silakan pilih Hapus.
11. (Opsional) Jika Anda ingin alarm melakukan tindakan untuk Amazon EC2 Auto Scaling, Amazon EC2, tiket, AWS Systems Manager atau, pilih tombol yang sesuai, dan tentukan status dan tindakan alarm.

> **Note**
>
> Alarm Anda dapat melakukan tindakan Systems Manager hanya ketika alarm tersebut berada dalam status `ALARM`. Untuk informasi tentang tindakan Systems Manager, lihat Mengkonfigurasi CloudWatch untuk membuat OpsItems dan Pembuatan insiden.

12. Pilih Berikutnya.
13. Pada Nama dan deskripsi, Anda harus memasukkan nama dan deskripsi untuk alarm Anda, dan kemudian pilih Berikutnya. Nama tersebut harus menggunakan karakter UTF-8, dan tidak dapat berisi karakter kontrol ASCII. Deskripsi dapat mencakup pemformatan penurunan harga, yang hanya ditampilkan di tab Detail alarm di CloudWatch konsol. Penurunan harga dapat Anda gunakan untuk menambahkan tautan ke runbook atau sumber daya internal lainnya.

> **Tip**
>
> Nama alarm harus menggunakan karakter UTF-8 saja. Nama tersebut tidak boleh memuat karakter kontrol ASCII.

14. Pada Pratinjau dan buat, silakan Anda konfirmasi bahwa informasi dan kondisi alarm Anda sudah benar, dan kemudian pilih Buat alarm.

Metrik yang diterbitkan oleh detektor anomali log

CloudWatch Log menerbitkan AnomalyCountmetrik ke CloudWatch metrik. Metrik ini dipublikasikan ke `AWS/Logs` namespace.

AnomalyCountMetrik diterbitkan dengan dimensi berikut:

- LogAnomalyDetector— Nama detektor anomali
- LogAnomalyPriority— Tingkat prioritas anomali

Enkripsi detektor anomali dan hasilnya dengan AWS KMS

Data detektor anomali selalu dienkripsi di Log. CloudWatch Secara default, CloudWatch Log menggunakan enkripsi sisi server untuk data saat istirahat. Sebagai alternatif, Anda dapat menggunakan AWS Key Management Service enkripsi ini. Jika Anda melakukannya, enkripsi

dilakukan dengan menggunakan AWS KMS kunci. Penggunaan enkripsi AWS KMS diaktifkan pada tingkat detektor anomali, dengan mengaitkan kunci KMS dengan detektor anomali.

> **Important**
>
> CloudWatch Log hanya mendukung kunci KMS simetris. Jangan gunakan kunci asimetris untuk mengenkripsi data dalam grup log Anda. Untuk informasi selengkapnya, lihat Menggunakan Kunci Simetris dan Asimetris.

Batas

- Untuk melakukan langkah-langkah berikut, Anda harus memiliki izin berikut: `kms:CreateKey`, `kms:GetKeyPolicy`, dan `kms:PutKeyPolicy`.
- Setelah Anda mengaitkan atau melepaskan kunci dari detektor anomali, diperlukan waktu hingga lima menit agar operasi diterapkan.
- Jika Anda mencabut akses CloudWatch Log ke kunci terkait atau menghapus kunci KMS terkait, data terenkripsi Anda di CloudWatch Log tidak dapat diambil lagi.

Langkah 1: Buat AWS KMS kunci

Untuk membuat kunci KMS, gunakan perintah create-key berikut:

```
aws kms create-key
```

Output berisi ID kunci dan Amazon Resource Name (ARN) dari kunci. Berikut ini adalah output contoh:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "key-default-1",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
```

```
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/key-default-1",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Langkah 2: Tetapkan izin pada tombol KMS

Secara default, semua AWS KMS kunci bersifat pribadi. Hanya pemilik sumber daya yang dapat menggunakannya untuk mengenkripsi dan mendekripsi data. Namun, pemilik sumber daya dapat memberikan izin untuk mengakses kunci KMS ke pengguna dan sumber daya lain. Dengan langkah ini, Anda memberikan izin utama layanan CloudWatch Log untuk menggunakan kunci. Prinsipal layanan ini harus berada di AWS Wilayah yang sama di mana kunci KMS disimpan.

Sebagai praktik terbaik, kami menyarankan Anda membatasi penggunaan kunci KMS hanya untuk AWS akun atau detektor anomali yang Anda tentukan.

Pertama, simpan kebijakan default untuk kunci KMS Anda seperti `policy.json` menggunakan get-key-policyperintah berikut:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Buka file `policy.json` di editor teks dan tambahkan bagian dalam huruf tebal dari salah satu pernyataan berikut. Pisahkan pernyataan yang ada dari pernyataan baru dengan koma. Pernyataan ini menggunakan `Condition` bagian untuk meningkatkan keamanan AWS KMS kunci. Untuk informasi selengkapnya, lihat AWS KMS kunci dan konteks enkripsi.

`Condition`Bagian dalam contoh ini membatasi penggunaan AWS KMS kunci ke akun yang ditentukan, tetapi dapat digunakan untuk detektor anomali apa pun.

```
{
  "Version": "2012-10-17",
  "Id": "key-default-1",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
```

```
        "AWS": "arn:aws:iam::Your_account_ID:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.REGION.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "kms:EncryptionContext:aws:logs:arn":
"arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
        }
      }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.REGION.amazonaws.com"
        },
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
        ],
        "Resource": "*",
        "Condition": {
            "ArnLike": {
                "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
"arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
            }
        }
    }
```

```
  ]
}
```

Terakhir, tambahkan kebijakan yang diperbarui menggunakan put-key-policyperintah berikut:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Langkah 3: Kaitkan kunci KMS dengan detektor anomali

Anda dapat mengaitkan kunci KMS dengan detektor anomali saat membuatnya di konsol atau menggunakan API atau. AWS CLI

Langkah 4: Lepaskan kunci dari detektor anomali

Setelah kunci dikaitkan dengan detektor anomali, Anda tidak dapat memperbarui kunci. Satu-satunya cara untuk menghapus kunci adalah dengan menghapus detektor anomali, dan kemudian membuatnya kembali.

Bekerja dengan grup log dan pengaliran log

Pengaliran log adalah urutan log acara yang berbagi sumber yang sama. Setiap sumber log yang terpisah di CloudWatch Log membentuk aliran log terpisah.

Grup log adalah grup pengaliran log yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama. Anda dapat menentukan grup log dan menentukan pengaliran untuk dimasukkan ke dalam setiap grup. Tidak ada batas jumlah pengaliran log yang dapat tergabung dalam satu grup log.

Gunakan prosedur di bagian ini untuk bekerja dengan grup log dan pengaliran log.

Buat grup log di CloudWatch Log

Saat Anda menginstal agen CloudWatch Log di instans Amazon EC2 menggunakan langkah-langkah di bagian sebelumnya dari Panduan Pengguna Amazon CloudWatch Logs, grup log dibuat sebagai bagian dari proses tersebut. Anda juga dapat membuat grup log langsung di CloudWatch konsol.

Untuk membuat grup log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Pilih Actions (Tindakan), lalu pilih Create log group (Buat grup log).
4. Masukkan nama untuk grup log, lalu pilih Create log group (Buat grup log).

> **Tip**
>
> Anda dapat grup log favorit, serta dasbor dan alarm, dari menu Favorit dan terbaru di panel navigasi. Di bawah kolom Baru dikunjungi, arahkan kursor ke grup log yang ingin Anda sukai, dan pilih simbol bintang di sebelahnya.

Mengirim log ke grup log

CloudWatch Log secara otomatis menerima peristiwa log dari beberapa AWS layanan. Anda juga dapat mengirim peristiwa log lainnya ke CloudWatch Log menggunakan salah satu metode berikut:

- CloudWatch agen — CloudWatch Agen terpadu dapat mengirim metrik dan log ke CloudWatch Log. Untuk informasi tentang menginstal dan menggunakan CloudWatch agen, lihat Mengumpulkan Metrik dan Log dari Instans Amazon EC2 dan Server Lokal dengan CloudWatch Agen di Panduan Pengguna Amazon. CloudWatch
- AWS CLIput-log-events—Mengunggah kumpulan peristiwa log ke Log. CloudWatch
- Secara terprogram - PutLogEventsAPI memungkinkan Anda untuk mengunggah batch peristiwa log secara terprogram ke Log. CloudWatch

Lihat data log yang dikirim ke CloudWatch Log

Anda dapat melihat dan menggulir data log stream-by-stream berdasarkan yang dikirim ke CloudWatch Log oleh agen CloudWatch Log. Anda dapat menentukan rentang waktu untuk data log yang akan dilihat.

Untuk melihat data log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Untuk Log Groups (Grup Log), pilih grup log untuk melihat pengaliran.
4. Dalam daftar grup log, pilih nama grup log yang ingin Anda lihat.
5. Dalam daftar pengaliran log, pilih nama pengaliran log yang ingin Anda lihat.
6. Untuk mengubah cara data log ditampilkan, lakukan salah satu hal berikut:
   - Untuk memperluas satu log acara, pilih tanda panah di samping log acara tersebut.
   - Untuk memperluas semua log acara dan melihatnya sebagai teks biasa, di atas daftar log acara, pilih Text (Teks).
   - Untuk memfilter log acara, masukkan filter pencarian yang diinginkan di kolom pencarian. Untuk informasi selengkapnya, lihat Membuat metrik dari peristiwa log menggunakan filter.
   - Untuk melihat data log untuk tanggal dan rentang waktu yang ditentukan, di samping filter pencarian, pilih tanda panah di samping tanggal dan waktu. Untuk menentukan rentang tanggal dan waktu, pilih Absolute (Absolut). Untuk memilih jumlah menit, jam, hari, atau minggu yang telah ditentukan, pilih Relative (Relatif). Anda juga dapat beralih antara UTC dan zona waktu lokal.

Gunakan Live Tail untuk melihat log dalam waktu dekat

CloudWatch Logs Live Tail membantu Anda memecahkan masalah insiden dengan cepat dengan melihat daftar streaming peristiwa log baru saat tertelan. Anda dapat melihat, memfilter, dan menyorot log yang dicerna dalam waktu dekat, membantu Anda mendeteksi dan menyelesaikan masalah dengan cepat. Anda dapat memfilter log berdasarkan istilah yang Anda tentukan, dan juga menyorot log yang berisi istilah tertentu untuk membantu Anda menemukan apa yang Anda cari dengan cepat.

Sesi Live Tail dikenakan biaya berdasarkan waktu penggunaan sesi, per menit. Untuk informasi selengkapnya tentang harga, lihat tab Log di CloudWatch Harga Amazon.

> **Note**
>
> Live Tail hanya didukung untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

Bagian berikut menjelaskan cara menggunakan Live Tail di konsol. Anda juga dapat memulai sesi Live Tail secara terprogram. Untuk informasi lebih lanjut, lihat StartLiveTail. Untuk contoh SDK, lihat Memulai sesi Live Tail menggunakan AWS SDK.

Memulai sesi Live Tail

Anda menggunakan CloudWatch konsol untuk memulai sesi Live Tail. Prosedur berikut menjelaskan cara memulai sesi Live Tail dengan menggunakan opsi Live tail di panel navigasi kiri. Anda juga dapat memulai sesi Live Tail dari halaman Grup Log atau halaman Wawasan CloudWatch Log.

> **Note**
>
> Jika Anda menggunakan kebijakan perlindungan data untuk menutupi data sensitif dalam grup log yang Anda lihat dengan Live Tail, data sensitif akan selalu muncul bertopeng di sesi Live Tail. Untuk informasi selengkapnya tentang menyembunyikan data sensitif di grup log, lihatMembantu melindungi data log sensitif dengan masking.

Untuk memulai sesi Live Tail

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

2. Di panel navigasi, pilih Log, Ekor langsung.
3. Untuk Pilih grup log, pilih grup log tempat Anda ingin melihat peristiwa, di sesi Live Tail. Anda dapat memilih sebanyak 10 grup log.
4. (Opsional) Jika Anda memilih hanya satu grup log, Anda dapat memfilter sesi Live Tail Anda lebih lanjut dengan memilih satu atau beberapa aliran log untuk melihat peristiwa log. Untuk melakukannya, di bawah Pilih aliran log, pilih nama aliran log dari daftar drop-down. Atau, Anda dapat menggunakan kotak kedua di bawah Pilih aliran log untuk memasukkan awalan nama aliran log, dan kemudian semua aliran log dengan nama yang cocok dengan awalan akan dipilih.
5. (Opsional) Untuk menampilkan hanya peristiwa log yang berisi kata-kata tertentu atau string lainnya, masukkan kata atau string di`Add filter patterns`.

   Misalnya, untuk menampilkan hanya peristiwa log yang menyertakan kata`Warning`, masukkan**`Warning`**. Bidang filter peka huruf besar/kecil. Anda dapat menyertakan beberapa operator istilah dan pola di bidang ini:

   - **`error 404`**hanya menampilkan peristiwa log yang mencakup keduanya `error` dan `404`
   - **`?Error ?error`**menampilkan peristiwa log yang mencakup salah satu `Error` atau `error`
   - **`-INFO`**menampilkan semua peristiwa log yang tidak termasuk `INFO`
   - **`{ $.eventType = "UpdateTrail" }`**menampilkan semua peristiwa log JSON di mana nilai bidang jenis acara `UpdateTrail`

   Anda juga dapat menggunakan ekspresi reguler (regex) untuk memfilter:

   - **`%ERROR%`**menggunakan regex untuk menampilkan semua peristiwa log yang terdiri dari kata kunci ERROR
   - **`{ $.names = %Steve% }`**menggunakan regex untuk menampilkan peristiwa log JSON di mana Steve berada di properti `"name"`
   - **`[ w1 = %abc%, w2 ]`**menggunakan regex untuk menampilkan peristiwa log yang dibatasi ruang di mana kata pertama adalah abc

   Untuk informasi selengkapnya tentang sintaks pola, lihat Filter sintaks pola.
6. (0ptional) Untuk menyorot beberapa peristiwa log yang ditampilkan, masukkan istilah untuk dicari dan sorot di bawah Live Tail. Masukkan istilah sorotan satu per satu. Jika Anda menambahkan beberapa istilah untuk disorot, warna yang berbeda ditetapkan untuk mewakili setiap istilah. Indikator sorotan ditampilkan di sebelah kiri setiap peristiwa log yang berisi istilah yang

ditentukan, dan juga muncul di bawah istilah itu sendiri ketika Anda memperluas peristiwa log di jendela utama untuk melihat peristiwa log lengkap.

Anda dapat menggunakan pemfilteran bersama dengan penyorotan untuk memecahkan masalah dengan cepat. Misalnya, Anda dapat memfilter peristiwa untuk menampilkan hanya peristiwa yang berisi`Error`, dan kemudian juga menyorot peristiwa yang berisi`404`.

7. Untuk memulai sesi, pilih Terapkan filter

   Peristiwa log yang cocok mulai muncul di jendela. Informasi berikut juga ditampilkan:

   - Timer menampilkan berapa lama sesi Live Tail telah aktif.
   - acara/detik menampilkan berapa banyak peristiwa log tertelan per detik yang cocok dengan filter yang telah Anda tetapkan.
   - Agar sesi tidak bergulir terlalu cepat karena banyak acara cocok dengan filter, CloudWatch Log mungkin hanya menampilkan beberapa peristiwa yang cocok. Jika ini terjadi, persentase peristiwa pencocokan yang ditampilkan di layar ditampilkan dalam% ditampilkan.
8. Untuk menjeda alur peristiwa untuk menyelidiki apa yang saat ini ditampilkan, klik di mana saja di jendela peristiwa.
9. Selama sesi, Anda dapat menggunakan yang berikut ini untuk melihat detail lebih lanjut tentang setiap peristiwa log.

   - Untuk menampilkan seluruh teks untuk peristiwa log di jendela utama, pilih panah di sebelah peristiwa log itu.
   - Untuk menampilkan seluruh teks untuk peristiwa log di jendela samping, pilih kaca pembesar + di sebelah peristiwa log itu. Alur acara berhenti dan jendela samping muncul.

     Menampilkan teks peristiwa log di jendela samping dapat berguna untuk membandingkan teksnya dengan peristiwa lain di jendela utama.
10. Untuk menghentikan sesi Live Tail, pilih Stop.
11. Untuk memulai ulang sesi, secara opsional gunakan panel Filter untuk memodifikasi kriteria pemfilteran, dan pilih Terapkan filter. Kemudian pilih Mulai.

Cari data log menggunakan pola filter

Anda dapat mencari data log Anda menggunakan Filter sintaks pola untuk filter metrik, filter langganan, peristiwa log filter, dan Live Tail. Anda dapat mencari semua aliran log dalam grup

log, atau dengan menggunakan AWS CLI Anda juga dapat mencari aliran log tertentu. Saat pencarian berjalan, akan dihasilkan halaman pertama data yang ditemukan dan token untuk mengambil halaman berikutnya dari data atau untuk melanjutkan pencarian. Jika tidak ada hasil yang dikembalikan, Anda dapat melanjutkan pencarian.

Anda dapat mengatur rentang waktu yang ingin Anda kuerikan untuk membatasi cakupan pencarian Anda. Anda bisa mulai dengan rentang yang lebih besar untuk melihat tempat garis log yang Anda inginkan, dan kemudian mempersingkat rentang waktu untuk membuat cakupan tampilan log dalam rentang waktu yang Anda inginkan.

Anda juga dapat beralih langsung dari metrik yang diekstraksi log ke log yang sesuai.

Jika Anda masuk ke akun yang disiapkan sebagai akun pemantauan dalam pengamatan CloudWatch lintas akun, Anda dapat mencari dan memfilter peristiwa log dari akun sumber yang ditautkan ke akun pemantauan ini. Untuk informasi lebih lanjut, lihat CloudWatch observabilitas lintas akun.

Cari entri log menggunakan konsol

Anda dapat mencari entri log yang memenuhi kriteria tertentu menggunakan konsol.

Untuk mencari log menggunakan konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Untuk Log Groups (Grup Log), pilih nama grup log yang berisi pengaliran log yang akan dicari.
4. Untuk Log Stream, pilih nama log stream yang akan dicari.
5. Di bawah Log events (Log acara), masukkan sintaks filter yang akan digunakan.

Untuk mencari semua entri log untuk rentang waktu menggunakan konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Untuk Log Groups (Grup Log), pilih nama grup log yang berisi pengaliran log yang akan dicari.
4. Pilih Search log group (Cari grup log).
5. Untuk Log events (Log acara), pilih tanggal dan rentang waktu, dan masukkan sintaks filter.

Cari entri log menggunakan AWS CLI

Anda dapat mencari entri log yang memenuhi kriteria tertentu menggunakan. AWS CLI

Untuk mencari entri log menggunakan AWS CLI

Pada prompt perintah, jalankan filter-log-eventsperintah berikut. Gunakan `--filter-pattern` untuk membatasi hasil ke pola filter yang ditentukan dan `--log-stream-names` untuk membatasi hasil ke pengaliran log tertentu.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Untuk mencari entri log selama rentang waktu tertentu menggunakan AWS CLI

Pada prompt perintah, jalankan filter-log-eventsperintah berikut:

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-
time 1482217558365] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Pivot dari metrik ke log

Anda bisa beralih ke entri log tertentu dari bagian lain dari konsol.

Untuk beralih dari widget dasbor ke log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, silakan pilih Dasbor.
3. Pilih dasbor.
4. Di widget, pilih ikon View logs (Lihat log), lalu pilih View logs in this time range (Lihat log dalam rentang waktu ini). Jika terdapat lebih dari satu filter metrik, pilih salah satu dari daftar. Jika ada lebih banyak filter metrik dari yang dapat kita tampilkan dalam daftar, pilih More metric filters (Lebih banyak filter metrik) dan pilih atau cari filter metrik.

Untuk beralih dari metrik ke log

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

2. Pada panel navigasi, silakan pilih Metrik.
3. Di bidang pencarian di tab All metrics, ketik nama metrik dan tekan Enter.
4. Pilih satu atau beberapa metrik dari hasil pencarian Anda.
5. Pilih Actions (Tindakan), View logs (Lihat log). Jika terdapat lebih dari satu filter metrik, pilih salah satu dari daftar. Jika ada lebih banyak filter metrik dari yang dapat kita tampilkan dalam daftar, pilih More metric filters (Lebih banyak filter metrik) dan pilih atau cari filter metrik.

Pemecahan Masalah

Pencarian membutuhkan waktu terlalu lama untuk diselesaikan

Jika Anda memiliki banyak data log, pencarian mungkin memerlukan waktu lama untuk diselesaikan. Untuk mempercepat pencarian, Anda dapat melakukan hal berikut:

- Jika Anda menggunakan AWS CLI, Anda dapat membatasi pencarian hanya pada aliran log yang Anda minati. Misalnya, jika grup log Anda memiliki 1000 aliran log, tetapi Anda hanya ingin melihat tiga aliran log yang Anda tahu relevan, Anda dapat menggunakan AWS CLI untuk membatasi pencarian Anda hanya pada tiga aliran log dalam grup log.
- Gunakan rentang waktu yang lebih pendek dan lebih terperinci, yang mengurangi jumlah data yang akan dicari dan mempercepat kueri.

Ubah penyimpanan data log di CloudWatch Log

Secara default, data log disimpan di CloudWatch Log tanpa batas waktu. Namun, Anda dapat mengonfigurasi berapa lama data log disimpan dalam grup log. Data apa pun yang lebih lama dari pengaturan retensi saat ini akan dihapus. Anda dapat mengubah retensi log untuk setiap grup log kapan saja.

> **Note**
>
> CloudWatch Logs tidak segera menghapus peristiwa log ketika mereka mencapai pengaturan retensi mereka. Biasanya memakan waktu hingga 72 jam setelah itu sebelum peristiwa log dihapus, tetapi dalam situasi yang jarang terjadi mungkin memakan waktu lebih lama.
> Ini berarti bahwa jika Anda mengubah grup log untuk memiliki pengaturan retensi yang lebih lama ketika berisi peristiwa log yang melewati tanggal kedaluwarsa, tetapi belum benar-benar dihapus, peristiwa log tersebut akan memakan waktu hingga 72 jam untuk dihapus

setelah tanggal penyimpanan baru tercapai. Untuk memastikan bahwa data log dihapus secara permanen, simpan grup log pada pengaturan retensi yang lebih rendah hingga 72 jam berlalu setelah akhir periode penyimpanan sebelumnya, atau Anda telah mengonfirmasi bahwa peristiwa log lama akan dihapus.
Ketika peristiwa log mencapai pengaturan retensi mereka, mereka ditandai untuk dihapus. Setelah ditandai untuk dihapus, mereka tidak menambah biaya penyimpanan arsip Anda lagi, bahkan jika mereka tidak benar-benar dihapus sampai nanti. Peristiwa log yang ditandai untuk dihapus ini juga tidak disertakan saat Anda menggunakan API untuk mengambil `storedBytes` nilai guna melihat berapa banyak byte yang disimpan grup log.

Untuk mengubah pengaturan retensi log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, Grup log.
3. Temukan grup log yang akan diperbarui.
4. Di kolom Retensi untuk grup log tersebut, pilih pengaturan retensi saat ini, seperti Jangan Pernah Kedaluwarsa.
5. Di Setelan retensi, untuk peristiwa kedaluwarsa setelahnya, pilih nilai retensi log, lalu pilih Simpan.

Tandai grup log di Amazon CloudWatch Logs

Anda dapat menetapkan metadata Anda sendiri ke grup log yang Anda buat di Amazon CloudWatch Logs dalam bentuk tag. Tanda adalah pasangan nilai-kunci yang Anda tetapkan untuk grup log. Menggunakan tag adalah cara sederhana namun ampuh untuk mengelola AWS sumber daya dan mengatur data, termasuk data penagihan.

Note

Anda dapat menggunakan tag untuk mengontrol akses ke sumber CloudWatch Log, termasuk grup log dan tujuan. Akses ke aliran log dikontrol pada tingkat grup log, karena hubungan hierarkis antara grup log dan aliran log. Untuk informasi selengkapnya tentang penggunaan tanda untuk mengendalikan akses, lihat Mengendalikan akses ke sumber daya Amazon Web Services menggunakan tanda.

Daftar Isi

Dasar-dasar tanda

Anda menggunakan AWS CloudFormation AWS CLI, atau CloudWatch Logs API untuk menyelesaikan tugas-tugas berikut:

- Menambahkan tanda ke grup log saat Anda membuatnya.
- Menambahkan tanda ke grup log yang sudah ada.
- Mendaftar tanda untuk grup log.
- Menghapus tanda dari grup log.

Anda dapat menggunakan tanda untuk mengategorikan grup log Anda. Misalnya, Anda dapat mengategorikannya berdasarkan tujuan, pemilik, atau lingkungan. Karena Anda menentukan kunci dan nilai untuk setiap tanda, Anda dapat membuat serangkaian kategori khusus untuk memenuhi kebutuhan spesifik Anda. Misalnya, Anda dapat menentukan satu set tanda yang membantu Anda melacak grup log berdasarkan pemilik dan aplikasi terkait. Berikut adalah beberapa contoh tanda:

- Proyek: Nama proyek
- Pemilik: Nama
- Tujuan: Pengujian beban
- Aplikasi: Nama aplikasi
- Lingkungan: Produksi

Melacak biaya menggunakan penandaan

Anda dapat menggunakan tag untuk mengkategorikan dan melacak biaya Anda AWS . Saat Anda menerapkan tag ke AWS sumber daya Anda, termasuk grup log, laporan alokasi AWS biaya Anda

mencakup penggunaan dan biaya yang dikumpulkan berdasarkan tag. Anda dapat menerapkan tag yang mewakili kategori bisnis (seperti pusat biaya, nama aplikasi, atau pemilik) untuk mengatur biaya Anda di berbagai layanan. Untuk informasi selengkapnya, lihat Menggunakan Tanda Alokasi Biaya untuk Laporan Penagihan Khusus dalam Panduan Pengguna AWS Billing .

Batasan tag

Batasan berikut berlaku untuk tanda.

Batasan dasar

- Jumlah maksimum tanda per grup log adalah 50.
- Kunci dan nilai tag peka huruf besar dan kecil.
- Anda tidak dapat mengubah atau mengedit tanda untuk grup log yang dihapus.

Batasan kunci tanda

- Setiap kunci tanda harus unik. Jika Anda menambahkan tanda dengan kunci yang sudah digunakan, tanda baru akan menimpa pasangan nilai-kunci yang sudah ada.
- Anda tidak dapat memulai kunci tag `aws:` karena awalan ini dicadangkan untuk digunakan oleh AWS. AWS membuat tag yang dimulai dengan awalan ini atas nama Anda, tetapi Anda tidak dapat mengedit atau menghapusnya.
- Kunci tanda harus memiliki panjang antara 1 dan 128 karakter Unicode.
- Kunci tanda harus terdiri dari karakter berikut: huruf Unicode, digit, spasi, dan karakter khusus berikut: `_ . / = + - @`.

Batasan nilai tanda

- Panjang nilai tanda harus antara 0 dan 255 karakter Unicode.
- Nilai tanda dapat kosong. Jika tidak, nilai tanda harus terdiri dari karakter berikut: huruf Unicode, digit, spasi, dan salah satu karakter khusus berikut: `_ . / = + - @`.

Menandai grup log menggunakan AWS CLI

Anda dapat menambahkan, mendaftar, dan menghapus tanda menggunakan AWS CLI. Untuk contoh, lihat dokumentasi berikut:

create-log-group

Membuat grup log. Anda dapat secara opsional menambahkan tanda ketika membuat grup log.

tag-sumber daya

Menetapkan satu atau beberapa tag (pasangan kunci-nilai) ke sumber Log yang ditentukan CloudWatch .

list-tags-for-resource

Menampilkan tag yang terkait dengan sumber daya CloudWatch Log.

untag-sumber daya

Menghapus satu atau beberapa tag dari sumber CloudWatch Log yang ditentukan.

Menandai grup log menggunakan API CloudWatch Log

Anda dapat menambahkan, membuat daftar, dan menghapus tag menggunakan API CloudWatch Log. Untuk contoh, lihat dokumentasi berikut:

CreateLogGroup

Membuat grup log. Anda dapat secara opsional menambahkan tanda ketika membuat grup log.

TagResource

Menetapkan satu atau beberapa tag (pasangan kunci-nilai) ke sumber Log yang ditentukan CloudWatch .

ListTagsForResource

Menampilkan tag yang terkait dengan sumber daya CloudWatch Log.

UntagResource

Menghapus satu atau beberapa tag dari sumber CloudWatch Log yang ditentukan.

Enkripsi data log di CloudWatch Log menggunakan AWS Key Management Service

Data grup log selalu dienkripsi di CloudWatch Log. Secara default, CloudWatch Log menggunakan enkripsi sisi server untuk data log saat istirahat. Sebagai alternatif, Anda dapat menggunakan

AWS Key Management Service enkripsi ini. Jika Anda melakukannya, enkripsi dilakukan dengan menggunakan AWS KMS kunci. Penggunaan enkripsi AWS KMS diaktifkan pada tingkat grup log, dengan mengaitkan kunci KMS dengan grup log, baik saat Anda membuat grup log atau setelah ada.

> **Important**
>
> CloudWatch Log sekarang mendukung konteks enkripsi, menggunakan `kms:EncryptionContext:aws:logs:arn` sebagai kunci dan ARN dari grup log sebagai nilai untuk kunci itu. Jika Anda memiliki grup log yang telah dienkripsi dengan kunci KMS, dan Anda ingin membatasi kunci yang akan digunakan dengan satu akun dan grup log, Anda harus menetapkan kunci KMS baru yang menyertakan kondisi dalam kebijakan IAM. Untuk informasi selengkapnya, lihat AWS KMS kunci dan konteks enkripsi.

Setelah Anda mengaitkan kunci KMS dengan grup log, semua data yang baru dicerna untuk grup log dienkripsi menggunakan kunci ini. Data ini disimpan dalam format terenkripsi selama periode retensi. CloudWatch Log mendekripsi data ini setiap kali diminta. CloudWatch Log harus memiliki izin untuk kunci KMS setiap kali data terenkripsi diminta.

Jika Anda kemudian memisahkan kunci KMS dari grup CloudWatch log, Log mengenkripsi data yang baru dicerna menggunakan metode enkripsi default Log. CloudWatch Semua data yang dicerna sebelumnya yang dienkripsi dengan kunci KMS tetap dienkripsi dengan kunci KMS. CloudWatch Log masih dapat mengembalikan data tersebut setelah kunci KMS dipisahkan, karena CloudWatch Log masih dapat terus mereferensikan kunci tersebut. Namun, jika kunci kemudian dinonaktifkan, maka CloudWatch Log tidak dapat membaca log yang dienkripsi dengan kunci itu.

> **Important**
>
> CloudWatch Log hanya mendukung kunci KMS simetris. Jangan gunakan kunci asimetris untuk mengenkripsi data dalam grup log Anda. Untuk informasi selengkapnya, lihat Menggunakan Kunci Simetris dan Asimetris.

Batas

- Untuk melakukan langkah-langkah berikut, Anda harus memiliki izin berikut: `kms:CreateKey`, `kms:GetKeyPolicy`, dan `kms:PutKeyPolicy`.

- Setelah Anda mengaitkan atau memisahkan kunci dari grup log, diperlukan waktu hingga lima menit agar operasi diterapkan.
- Jika Anda mencabut akses CloudWatch Log ke kunci terkait atau menghapus kunci KMS terkait, data terenkripsi Anda di CloudWatch Log tidak dapat diambil lagi.
- Anda tidak dapat mengaitkan kunci KMS dengan grup log menggunakan CloudWatch konsol.

Langkah 1: Buat AWS KMS kunci

Untuk membuat kunci KMS, gunakan perintah create-key berikut:

```
aws kms create-key
```

Output berisi ID kunci dan Amazon Resource Name (ARN) dari kunci. Berikut ini adalah output contoh:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Langkah 2: Tetapkan izin pada tombol KMS

Secara default, semua AWS KMS kunci bersifat pribadi. Hanya pemilik sumber daya yang dapat menggunakannya untuk mengenkripsi dan mendekripsi data. Namun, pemilik sumber daya dapat

memberikan izin untuk mengakses kunci KMS ke pengguna dan sumber daya lain. Dengan langkah ini, Anda memberikan izin utama layanan CloudWatch Log untuk menggunakan kunci. Prinsipal layanan ini harus berada di AWS Wilayah yang sama di mana kunci KMS disimpan.

Sebagai praktik terbaik, kami menyarankan Anda membatasi penggunaan kunci KMS hanya untuk AWS akun atau grup log yang Anda tentukan.

Pertama, simpan kebijakan default untuk kunci KMS Anda seperti `policy.json` menggunakan get-key-policyperintah berikut:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Buka file `policy.json` di editor teks dan tambahkan bagian dalam huruf tebal dari salah satu pernyataan berikut. Pisahkan pernyataan yang ada dari pernyataan baru dengan koma. Pernyataan ini menggunakan `Condition` bagian untuk meningkatkan keamanan AWS KMS kunci. Untuk informasi selengkapnya, lihat AWS KMS kunci dan konteks enkripsi.

Bagian `Condition` dalam contoh ini membatasi kunci pada satu ARN grup log.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
```

```
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:log-group:log-group-name"
                }
            }
        }
    ]
}
```

Bagian `Condition` dalam contoh ini membatasi penggunaan kunci AWS KMS pada akun tertentu, tetapi dapat digunakan untuk grup log apa pun.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:*"
                }
            }
        }
    ]
}
```

Terakhir, tambahkan kebijakan yang diperbarui menggunakan put-key-policyperintah berikut:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Langkah 3: Kaitkan kunci KMS dengan grup log

Anda dapat mengaitkan kunci KMS dengan grup log saat Anda membuatnya atau setelah itu ada.

Untuk mengetahui apakah grup log sudah memiliki kunci KMS yang terkait, gunakan describe-log-groupsperintah berikut:

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

Jika outputnya mencakup bidang `kmsKeyId`, grup log terkait dengan kunci yang ditampilkan untuk nilai bidang tersebut.

Untuk mengaitkan kunci KMS dengan grup log saat Anda membuatnya

Gunakan create-log-groupperintah sebagai berikut:

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

Untuk mengaitkan kunci KMS dengan grup log yang ada

Gunakan associate-kms-keyperintah sebagai berikut:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

Langkah 4: Pisahkan kunci dari grup log

Untuk memisahkan kunci KMS yang terkait dengan grup log, gunakan perintah berikut: disassociate-kms-key

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

AWS KMS kunci dan konteks enkripsi

Untuk meningkatkan keamanan AWS Key Management Service kunci Anda dan grup log terenkripsi Anda, CloudWatch Log sekarang menempatkan ARN grup log sebagai bagian dari konteks enkripsi yang digunakan untuk mengenkripsi data log Anda. Konteks enkripsi adalah seperangkat pasangan nilai-kunci yang digunakan sebagai data terautentikasi tambahan. Konteks enkripsi memungkinkan Anda menggunakan kondisi kebijakan IAM untuk membatasi akses ke AWS KMS kunci Anda berdasarkan AWS akun dan grup log. Untuk informasi selengkapnya, lihat Konteks enkripsi and Elemen Kebijakan JSON IAM: Syarat.

Kami menyarankan Anda menggunakan kunci KMS yang berbeda untuk setiap grup log terenkripsi Anda.

Jika Anda memiliki grup log yang Anda enkripsi sebelumnya dan sekarang ingin mengubah grup log untuk menggunakan kunci KMS baru yang hanya berfungsi untuk grup log itu, ikuti langkah-langkah ini.

Untuk mengonversi grup log terenkripsi untuk menggunakan kunci KMS dengan kebijakan yang membatasi grup log tersebut

1. Masukkan perintah berikut untuk menemukan ARN dari kunci grup log saat ini:

   ```
   aws logs describe-log-groups
   ```

   Outputnya mencakup baris berikut. Perhatikan ARN. Anda perlu menggunakannya di langkah 7.

   ```
   ...
   "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
   cdef-0123-456789abcdef"
   ...
   ```

2. Masukkan perintah berikut untuk membuat kunci KMS baru:

```
aws kms create-key
```

3. Masukkan perintah berikut untuk menyimpan kebijakan kunci baru ke file `policy.json`:

```
aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
policy.json
```

4. Gunakan editor teks untuk membuka `policy.json` dan menambahkan ekspresi `Condition` ke kebijakan:

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT-ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn":
 "arn:aws:logs:REGION:ACCOUNT-ID:log-
group:LOG-GROUP-NAME"
                }
```

```
            }
        }
    ]
}
```

5. Masukkan perintah berikut untuk menambahkan kebijakan yang diperbarui ke kunci KMS baru:

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. Masukkan perintah berikut untuk mengaitkan kebijakan dengan grup log Anda:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

CloudWatch Log sekarang mengenkripsi semua data baru menggunakan kunci baru.

7. Selanjutnya, cabut semua izin kecuali `Decrypt` dari kunci lama. Pertama, masukkan perintah berikut untuk mengambil kebijakan lama:

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text
 > ./policy.json
```

8. Gunakan editor teks untuk membuka `policy.json` dan hapus semua nilai dari daftar `Action`, kecuali untuk `kms:Decrypt*`

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
```

```
            "Action": [
                "kms:Decrypt*"
            ],
            "Resource": "*"
        }
    ]
}
```

9. Masukkan perintah berikut untuk menambahkan kebijakan yang diperbarui ke kunci lama:

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

Membantu melindungi data log sensitif dengan masking

Anda dapat membantu melindungi data sensitif yang dicerna oleh CloudWatch Log dengan menggunakan kebijakan perlindungan data grup log. Kebijakan ini memungkinkan Anda mengaudit dan menutupi data sensitif yang muncul dalam peristiwa log yang dicerna oleh grup log di akun Anda.

Saat Anda membuat kebijakan perlindungan data, maka secara default, data sensitif yang cocok dengan pengidentifikasi data yang Anda pilih akan disembunyikan di semua titik keluar, termasuk Wawasan CloudWatch Log, filter metrik, dan filter langganan. Hanya pengguna yang memiliki izin `logs:Unmask` IAM yang dapat melihat data yang dibuka kedoknya.

Anda dapat membuat kebijakan perlindungan data untuk semua grup log di akun Anda, dan Anda juga dapat membuat kebijakan perlindungan data untuk grup log individual. Saat Anda membuat kebijakan untuk seluruh akun, kebijakan tersebut berlaku untuk grup log dan grup log yang sudah ada yang dibuat di masa mendatang.

Jika Anda membuat kebijakan perlindungan data untuk seluruh akun Anda dan Anda juga membuat kebijakan untuk satu grup log, kedua kebijakan tersebut berlaku untuk grup log tersebut. Semua pengidentifikasi data terkelola yang ditentukan dalam salah satu kebijakan diaudit dan disamarkan dalam grup log tersebut.

> **Note**
>
> Menyembunyikan data sensitif hanya didukung untuk grup log di kelas log Standar. Jika Anda membuat kebijakan perlindungan data untuk semua grup log di akun Anda, kebijakan

tersebut hanya berlaku untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

Setiap grup log hanya dapat memiliki satu kebijakan perlindungan data tingkat grup log, tetapi kebijakan tersebut dapat menentukan banyak pengidentifikasi data terkelola untuk diaudit dan disembunyikan. Batas untuk kebijakan perlindungan data adalah 30.720 karakter.

Important

Data sensitif terdeteksi dan disamarkan saat tertelan ke dalam grup log. Saat Anda menetapkan kebijakan perlindungan data, peristiwa log yang dicerna ke grup log sebelum waktu tersebut tidak disamarkan.

CloudWatch Log mendukung banyak pengidentifikasi data terkelola, yang menawarkan tipe data yang telah dikonfigurasi sebelumnya yang dapat Anda pilih untuk melindungi data keuangan, informasi kesehatan pribadi (PHI), dan informasi identitas pribadi (PII). CloudWatch Perlindungan data log memungkinkan Anda memanfaatkan pencocokan pola dan model pembelajaran mesin untuk mendeteksi data sensitif. Untuk beberapa jenis pengidentifikasi data terkelola, deteksi tergantung pada juga menemukan kata kunci tertentu yang berdekatan dengan data sensitif. Anda juga dapat menggunakan pengidentifikasi data khusus untuk membuat pengidentifikasi data yang disesuaikan dengan kasus penggunaan spesifik Anda.

Metrik dipancarkan CloudWatch saat data sensitif terdeteksi yang cocok dengan pengidentifikasi data yang Anda pilih. Ini adalah LogEventsWithFindingsmetrik dan dipancarkan di ruang nama AWS/log. Anda dapat menggunakan metrik ini untuk membuat CloudWatch alarm, dan Anda dapat memvisualisasikannya dalam grafik dan dasbor. Metrik yang dipancarkan oleh perlindungan data adalah metrik yang dijual dan tidak dikenai biaya. Untuk informasi selengkapnya tentang metrik yang dikirimkan oleh CloudWatch Log CloudWatch, lihatPemantauan dengan CloudWatch metrik.

Setiap pengidentifikasi data terkelola dirancang untuk mendeteksi jenis data sensitif tertentu, seperti nomor kartu kredit, kunci akses AWS rahasia, atau nomor paspor untuk negara atau wilayah tertentu. Saat membuat kebijakan perlindungan data, Anda dapat mengonfigurasinya untuk menggunakan pengidentifikasi ini untuk menganalisis log yang dicerna oleh grup log, dan mengambil tindakan saat terdeteksi.

CloudWatch Perlindungan data log dapat mendeteksi kategori data sensitif berikut dengan menggunakan pengidentifikasi data terkelola:

- Kredensil, seperti kunci pribadi atau kunci akses AWS rahasia
- Informasi keuangan, seperti nomor kartu kredit
- Informasi Identifikasi Pribadi (PII) seperti SIM atau nomor jaminan sosial
- Informasi Kesehatan yang Dilindungi (PHI) seperti asuransi kesehatan atau nomor identifikasi medis
- Pengidentifikasi perangkat, seperti alamat IP atau alamat MAC

Untuk detail tentang jenis data yang dapat Anda lindungi, lihatJenis data yang dapat Anda lindungi.

Daftar Isi

Memahami kebijakan perlindungan data

Topik

Apa itu kebijakan perlindungan data?

CloudWatch Log menggunakan kebijakan perlindungan data untuk memilih data sensitif yang ingin Anda pindai, dan tindakan yang ingin Anda ambil untuk melindungi data tersebut. Untuk memilih data sensitif yang menarik, Anda menggunakan pengidentifikasi data. CloudWatch Perlindungan data log kemudian mendeteksi data sensitif dengan menggunakan pembelajaran mesin dan pencocokan pola. Untuk menindaklanjuti pengidentifikasi data yang ditemukan, Anda dapat menentukan operasi audit dan de-identifikasi. Operasi ini memungkinkan Anda mencatat data sensitif yang ditemukan (atau tidak ditemukan), dan untuk menutupi data sensitif saat peristiwa log dilihat.

Bagaimana kebijakan perlindungan data terstruktur?

Seperti yang diilustrasikan pada gambar berikut, dokumen kebijakan perlindungan data mencakup elemen-elemen berikut:

- Informasi opsional untuk seluruh kebijakan di bagian atas dokumen
- Satu pernyataan yang mendefinisikan tindakan audit dan de-identifikasi

Hanya satu kebijakan perlindungan data yang dapat ditentukan per grup CloudWatch log Log. Kebijakan perlindungan data dapat memiliki satu atau lebih pernyataan penolakan atau de-identifikasi, tetapi hanya satu pernyataan audit.

Properti JSON untuk kebijakan perlindungan data

Kebijakan perlindungan data memerlukan informasi kebijakan dasar berikut untuk identifikasi:

- Nama — Nama kebijakan.
- Deskripsi (Opsional) — Deskripsi kebijakan.
- Versi - Versi bahasa kebijakan. Versi saat ini adalah 2021-06-01.
- Pernyataan — Daftar pernyataan yang menentukan tindakan kebijakan perlindungan data.

```
{
  "Name": "CloudWatchLogs-PersonalInformation-Protection",
  "Description": "Protect basic types of sensitive data",
  "Version": "2021-06-01",
  "Statement": [
        ...
  ]
```

```
}
```

Properti JSON untuk pernyataan kebijakan

Pernyataan kebijakan menetapkan konteks deteksi untuk operasi perlindungan data.

- Sid (Opsional) - Pengidentifikasi pernyataan.
- DataIdentifier— Data sensitif yang harus dipindai oleh CloudWatch Log. Misalnya, nama, alamat, atau nomor telepon.
- Operasi — Tindakan tindak lanjut, baik Audit atau De-identifikasi. CloudWatch Log melakukan tindakan ini ketika menemukan data sensitif.

```
{
  ...
  "Statement": [
    {
      "Sid": "audit-policy",
      "DataIdentifier": [
        "arn:aws:dataprotection::aws:data-identifier/Address"
      ],
      "Operation": {
        "Audit": {
          "FindingsDestination": {}
        }
      }
    },
```

Properti JSON untuk operasi pernyataan kebijakan

Pernyataan kebijakan menetapkan salah satu operasi perlindungan data berikut.

- Audit — Memancarkan laporan metrik dan temuan tanpa mengganggu pencatatan. String yang cocok menambah LogEventsWithFindingsmetrik yang diterbitkan CloudWatch Log ke namespace AWS/Log. CloudWatch Anda dapat menggunakan metrik ini untuk membuat alarm.

  Untuk contoh laporan temuan, lihatLaporan temuan audit.

  Untuk informasi selengkapnya tentang metrik yang dikirimkan oleh CloudWatch Log CloudWatch, lihatPemantauan dengan CloudWatch metrik.
- De-identifikasi—Tutupi data sensitif tanpa mengganggu pencatatan.

Izin IAM diperlukan untuk membuat atau bekerja dengan kebijakan perlindungan data

Agar dapat bekerja dengan kebijakan perlindungan data untuk grup log, Anda harus memiliki izin tertentu seperti yang ditunjukkan pada tabel berikut. Izin berbeda untuk kebijakan perlindungan data di seluruh akun dan untuk kebijakan perlindungan data yang berlaku untuk satu grup log.

Izin yang diperlukan untuk kebijakan perlindungan data tingkat akun

> **Note**
>
> Jika Anda melakukan salah satu operasi ini di dalam fungsi Lambda, peran eksekusi Lambda dan batas izin juga harus menyertakan izin berikut.

Operasi	Izin IAM diperlukan	Sumber Daya
Membuat kebijakan perlindungan data tanpa tujuan audit	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
Membuat kebijakan perlindungan data dengan CloudWatch Log sebagai tujuan audit	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`logs:PutResourcePolicy`	*
	`logs:DescribeResourcePolicies`	*

Operasi	Izin IAM diperlukan	Sumber Daya
	`logs:DescribeLogGroups`	*
Membuat kebijakan perlindungan data dengan Firehose sebagai tujuan audit	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`firehose:TagDeliveryStream`	`arn:aws:logs:::deliverystream/` *YOUR_DELIVERY_STREAM*
Membuat kebijakan perlindungan data dengan Amazon S3 sebagai tujuan audit	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`s3:GetBucketPolicy`	`arn:aws:s3:::` *YOUR_BUCKET*
	`s3:PutBucketPolicy`	`arn:aws:s3:::` *YOUR_BUCKET*
Buka kedok peristiwa log bertopeng dalam grup log tertentu	`logs:Unmask`	`arn:aws:logs:::log-group:*`

Operasi	Izin IAM diperlukan	Sumber Daya
Melihat kebijakan perlindungan data yang ada	`logs:GetDataProtectionPolicy`	*
Menghapus kebijakan perlindungan data	`logs:DeleteAccountPolicy`	*
	`logs:DeleteDataProtectionPolicy`	*

Jika ada log audit perlindungan data yang sudah dikirim ke tujuan, maka kebijakan lain yang mengirim log ke tujuan yang sama hanya memerlukan izin `logs:PutDataProtectionPolicy` dan `logs:CreateLogDelivery` izin.

Izin yang diperlukan untuk kebijakan perlindungan data untuk satu grup log

> **Note**
>
> Jika Anda melakukan salah satu operasi ini di dalam fungsi Lambda, peran eksekusi Lambda dan batas izin juga harus menyertakan izin berikut.

Operasi	Izin IAM diperlukan	Sumber Daya
Membuat kebijakan perlindungan data tanpa tujuan audit	`logs:PutDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Membuat kebijakan perlindungan data dengan CloudWatch Log sebagai tujuan audit	`logs:PutDataProtectionPolicy`
	
	`logs:CreateLogDelivery`
	
	`logs:PutResourcePolicy`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		
		*
		
		*
		
		*

Operasi	Izin IAM diperlukan	Sumber Daya

	`logs:DescribeResourcePolicies`
	
	`logs:DescribeLogGroups`
		*

Membuat kebijakan perlindungan data dengan Firehose sebagai tujuan audit	`logs:PutDataProtectionPolicy`
	
	`logs:CreateLogDelivery`
	
	`firehose:TagDeliveryStream`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		
		*
		
		`arn:aws:logs:::deliverystream/` *YOUR_DELIVERY_STREAM*

Membuat kebijakan perlindungan data dengan Amazon S3 sebagai tujuan audit	`logs:PutDataProtectionPolicy`
	
	`logs:CreateLogDelivery`
	
	`s3:GetBucketPolicy`
	
	`s3:PutBucketPolicy`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		
		*
		
		`arn:aws:s3:::` *YOUR_BUCKET*
		
		`arn:aws:s3:::` *YOUR_BUCKET*

Buka kedok peristiwa log bertopeng	`logs:Unmask`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
Melihat kebijakan perlindungan data yang ada	`logs:GetDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Operasi	Izin IAM diperlukan	Sumber Daya
Menghapus kebijakan perlindungan data	`logs:DeleteDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Jika ada log audit perlindungan data yang sudah dikirim ke tujuan, maka kebijakan lain yang mengirim log ke tujuan yang sama hanya memerlukan izin `logs:PutDataProtectionPolicy` dan `logs:CreateLogDelivery` izin.

Contoh kebijakan perlindungan data

Contoh kebijakan berikut memungkinkan pengguna untuk membuat, melihat, dan menghapus kebijakan perlindungan data yang dapat mengirimkan temuan audit ke ketiga jenis tujuan audit. Itu tidak mengizinkan pengguna untuk melihat data yang dibuka kedoknya.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "YOUR_SID_1",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeLogGroups",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "YOUR_SID_2",
            "Effect": "Allow",
            "Action": [
                "logs:GetDataProtectionPolicy",
                "logs:DeleteDataProtectionPolicy",
                "logs:PutDataProtectionPolicy",
                "s3:PutBucketPolicy",
                "firehose:TagDeliveryStream",
                "s3:GetBucketPolicy"
            ],
```

```
            "Resource": [
                "arn:aws:firehose:::deliverystream/YOUR_DELIVERY_STREAM",
                "arn:aws:s3:::YOUR_BUCKET",
                "arn:aws:logs:::log-group:YOUR_LOG_GROUP:*"
            ]
        }
    ]
}
```

Buat kebijakan perlindungan data di seluruh akun

Anda dapat menggunakan konsol CloudWatch Log atau AWS CLI perintah untuk membuat kebijakan perlindungan data guna menutupi data sensitif untuk semua grup log di akun Anda. Melakukannya memengaruhi grup log saat ini dan grup log yang Anda buat di masa mendatang.

> **Important**
>
> Data sensitif terdeteksi dan disamarkan saat tertelan ke dalam grup log. Saat Anda menetapkan kebijakan perlindungan data, peristiwa log yang dicerna ke grup log sebelum waktu tersebut tidak disamarkan.

Topik

- Konsol
- AWS CLI

Konsol

Untuk menggunakan konsol untuk membuat kebijakan perlindungan data seluruh akun

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, silakan pilih Pengaturan. Itu terletak di dekat bagian bawah daftar.
3. Pilih tab Log.
4. Pilih Konfigurasikan
5. Untuk pengidentifikasi data terkelola, pilih jenis data yang ingin Anda audit dan tutupi untuk semua grup log Anda. Anda dapat mengetikkan kotak pilihan untuk menemukan pengidentifikasi yang Anda inginkan.

Kami menyarankan Anda hanya memilih pengenal data yang relevan untuk data log dan bisnis Anda. Memilih banyak jenis data dapat menyebabkan positif palsu.

Untuk detail tentang jenis data yang dapat Anda lindungi, lihatJenis data yang dapat Anda lindungi.

6. (Opsional) Jika Anda ingin mengaudit dan menutupi jenis data lain dengan menggunakan pengidentifikasi data khusus, pilih Tambahkan pengenal data khusus. Kemudian masukkan nama untuk tipe data dan ekspresi reguler yang akan digunakan untuk mencari jenis data tersebut dalam peristiwa log. Untuk informasi selengkapnya, lihat Pengidentifikasi data khusus.

   Kebijakan perlindungan data tunggal dapat mencakup hingga 10 pengidentifikasi data kustom. Setiap ekspresi reguler yang mendefinisikan pengenal data kustom harus 200 karakter atau kurang.

7. (Opsional) Pilih satu atau lebih layanan untuk mengirimkan temuan audit ke. Bahkan jika Anda memilih untuk tidak mengirim temuan audit ke salah satu layanan ini, tipe data sensitif yang Anda pilih akan tetap tertutup.

8. Pilih Aktifkan perlindungan data.

AWS CLI

Untuk menggunakan AWS CLI untuk membuat kebijakan perlindungan data

1. Gunakan editor teks untuk membuat file kebijakan bernama`DataProtectionPolicy.json`. Untuk informasi tentang sintaks kebijakan, lihat bagian berikut.

2. Masukkan perintah berikut:

```
aws logs put-account-policy \
--policy-name TEST_POLICY --policy-type "DATA_PROTECTION_POLICY" \
--policy-document file://policy.json \
--scope "ALL" \
--region us-west-2
```

Sintaks kebijakan perlindungan data untuk AWS CLI atau operasi API

Saat Anda membuat kebijakan perlindungan data JSON untuk digunakan dalam operasi AWS CLI perintah atau API, kebijakan tersebut harus menyertakan dua blok JSON:

- Blok pertama harus menyertakan `DataIdentifer` array dan `Operation` properti dengan `Audit` tindakan. `DataIdentifer`Array mencantumkan jenis data sensitif yang ingin Anda tutupi. Untuk informasi lebih lanjut tentang opsi yang tersedia, lihat Jenis data yang dapat Anda lindungi.

  `Operation`Properti dengan `Audit` tindakan diperlukan untuk menemukan istilah data sensitif. `Audit`Tindakan ini harus berisi `FindingsDestination` objek. Anda dapat menggunakan `FindingsDestination` objek tersebut secara opsional untuk mencantumkan satu atau beberapa tujuan untuk mengirim laporan temuan audit. Jika Anda menentukan tujuan seperti grup log, aliran Amazon Data Firehose, dan bucket S3, mereka harus sudah ada. Untuk contoh laporan audit findins, lihat. Laporan temuan audit
- Blok kedua harus menyertakan `DataIdentifer` array dan `Operation` properti dengan `Deidentify` tindakan. `DataIdentifer`Array harus sama persis dengan `DataIdentifer` array di blok pertama kebijakan.

  `Operation`Properti dengan `Deidentify` tindakan adalah apa yang sebenarnya menutupi data, dan itu harus berisi `"MaskConfig": {}` objek. `"MaskConfig": {}`Objek harus kosong.

Berikut ini adalah contoh kebijakan perlindungan data yang hanya menggunakan pengidentifikasi data terkelola. Kebijakan ini menutupi alamat email dan SIM Amerika Serikat.

Untuk informasi tentang kebijakan yang menentukan pengenal data kustom, lihatMenggunakan pengidentifikasi data khusus dalam kebijakan perlindungan data Anda.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
```

```
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Membuat kebijakan perlindungan data untuk satu grup log

Anda dapat menggunakan konsol CloudWatch Log atau AWS CLI perintah untuk membuat kebijakan perlindungan data untuk menutupi data sensitif.

Anda dapat menetapkan satu kebijakan perlindungan data untuk setiap grup log. Setiap kebijakan perlindungan data dapat mengaudit berbagai jenis informasi. Setiap kebijakan perlindungan data dapat mencakup satu pernyataan audit.

Topik

- Konsol
- AWS CLI

Konsol

Untuk menggunakan konsol untuk membuat kebijakan perlindungan data

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Log, Grup log.
3. Pilih nama grup log.
4. Pilih Tindakan, Buat kebijakan perlindungan data.
5. Untuk pengidentifikasi data terkelola, pilih jenis data yang ingin Anda audit dan tutupi di grup log ini. Anda dapat mengetikkan kotak pilihan untuk menemukan pengidentifikasi yang Anda inginkan.

   Kami menyarankan Anda hanya memilih pengenal data yang relevan untuk data log dan bisnis Anda. Memilih banyak jenis data dapat menyebabkan positif palsu.

   Untuk detail tentang jenis data yang dapat Anda lindungi dengan menggunakan pengenal data terkelola, lihatJenis data yang dapat Anda lindungi.
6. (Opsional) Jika Anda ingin mengaudit dan menutupi jenis data lain dengan menggunakan pengidentifikasi data khusus, pilih Tambahkan pengenal data khusus. Kemudian masukkan nama untuk tipe data dan ekspresi reguler yang akan digunakan untuk mencari jenis data tersebut dalam peristiwa log. Untuk informasi selengkapnya, lihat Pengidentifikasi data khusus.

   Kebijakan perlindungan data tunggal dapat mencakup hingga 10 pengidentifikasi data kustom. Setiap ekspresi reguler yang mendefinisikan pengenal data kustom harus 200 karakter atau kurang.
7. (Opsional) Pilih satu atau lebih layanan untuk mengirimkan temuan audit ke. Bahkan jika Anda memilih untuk tidak mengirim temuan audit ke salah satu layanan ini, tipe data sensitif yang Anda pilih akan tetap tertutup.
8. Pilih Aktifkan perlindungan data.

AWS CLI

Untuk menggunakan AWS CLI untuk membuat kebijakan perlindungan data

1. Gunakan editor teks untuk membuat file kebijakan bernama`DataProtectionPolicy.json`. Untuk informasi tentang sintaks kebijakan, lihat bagian berikut.
2. Masukkan perintah berikut:

```
aws logs put-data-protection-policy --log-group-identifier "my-log-group" --policy-
document file:///Path/DataProtectionPolicy.json --region us-west-2
```

Sintaks kebijakan perlindungan data untuk AWS CLI atau operasi API

Saat Anda membuat kebijakan perlindungan data JSON untuk digunakan dalam operasi AWS CLI perintah atau API, kebijakan tersebut harus menyertakan dua blok JSON:

- Blok pertama harus menyertakan `DataIdentifer` array dan `Operation` properti dengan `Audit` tindakan. `DataIdentifer`Array mencantumkan jenis data sensitif yang ingin Anda tutupi. Untuk informasi lebih lanjut tentang opsi yang tersedia, lihat Jenis data yang dapat Anda lindungi.

  `Operation`Properti dengan `Audit` tindakan diperlukan untuk menemukan istilah data sensitif. `Audit`Tindakan ini harus berisi `FindingsDestination` objek. Anda dapat menggunakan `FindingsDestination` objek tersebut secara opsional untuk mencantumkan satu atau beberapa tujuan untuk mengirim laporan temuan audit. Jika Anda menentukan tujuan seperti grup log, aliran Amazon Data Firehose, dan bucket S3, mereka harus sudah ada. Untuk contoh laporan audit findins, lihat. Laporan temuan audit
- Blok kedua harus menyertakan `DataIdentifer` array dan `Operation` properti dengan `Deidentify` tindakan. `DataIdentifer`Array harus sama persis dengan `DataIdentifer` array di blok pertama kebijakan.

  `Operation`Properti dengan `Deidentify` tindakan adalah apa yang sebenarnya menutupi data, dan itu harus berisi `"MaskConfig": {}` objek. `"MaskConfig": {}`Objek harus kosong.

Berikut ini adalah contoh kebijakan perlindungan data yang menutupi alamat email dan SIM Amerika Serikat.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
```

```
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Lihat data yang dibuka kedoknya

Untuk melihat data yang dibuka kedoknya, pengguna harus memiliki izin. `logs:Unmask` Pengguna dengan izin ini dapat melihat data yang dibuka kedoknya dengan cara berikut:

- Saat melihat peristiwa dalam aliran log, pilih Tampilan, Buka Kedok.
- Gunakan kueri CloudWatch Logs Insights yang menyertakan perintah unmask (@message). Contoh query berikut menampilkan 20 peristiwa log terbaru dalam aliran, membuka kedoknya:

```
fields @timestamp, @message, unmask(@message)
| sort @timestamp desc
```

```
| limit 20
```

Untuk informasi selengkapnya tentang perintah Wawasan CloudWatch Log, lihatCloudWatch Sintaks kueri Log Insights.

- Gunakan GetLogEventsatau FilterLogEventsoperasi dengan `unmask` parameter.

CloudWatchLogsFullAccessKebijakan tersebut termasuk `logs:Unmask` izin. Untuk memberikan `logs:Unmask` kepada pengguna yang tidak memiliki CloudWatchLogsFullAccess, Anda dapat melampirkan kebijakan IAM khusus kepada pengguna tersebut. Untuk informasi selengkapnya, lihat Menambahkan izin ke pengguna (konsol).

Laporan temuan audit

Jika Anda menyiapkan kebijakan audit perlindungan data CloudWatch Log untuk menulis laporan audit ke CloudWatch Log, Amazon S3, atau Firehose, laporan temuan ini serupa dengan contoh berikut. CloudWatch Log menulis satu laporan temuan untuk setiap peristiwa log yang berisi data sensitif.

```
{
    "auditTimestamp": "2023-01-23T21:11:20Z",
    "resourceArn": "arn:aws:logs:us-west-2:111122223333:log-group:/aws/lambda/
MyLogGroup:*",
    "dataIdentifiers": [
        {
            "name": "EmailAddress",
            "count": 2,
            "detections": [
                {
                    "start": 13,
                    "end": 26
                },
{
                    "start": 30,
                    "end": 43
                }
            ]
        }
    ]
}
```

Bidang dalam laporan adalah sebagai berikut:

- `resourceArn`Bidang menampilkan grup log tempat data sensitif ditemukan.
- `dataIdentifiers`Objek menampilkan informasi tentang temuan untuk satu jenis data senssitif yang Anda audit.
- `name`Bidang mengidentifikasi jenis data sensitif yang dilaporkan bagian ini.
- `count`Bidang menampilkan berapa kali jenis data sensitif ini muncul dalam peristiwa log.
- `end`Bidang `start` dan menunjukkan di mana dalam peristiwa log, berdasarkan jumlah karakter, setiap kemunculan data sensitif muncul.

Contoh sebelumnya menunjukkan laporan menemukan dua alamat email dalam satu peristiwa log. Alamat email pertama dimulai pada karakter ke-13 dari peristiwa log dan berakhir pada karakter ke-26. Alamat email kedua berjalan dari karakter ke-30 ke karakter ke-43. Meskipun peristiwa log ini memiliki dua alamat email, nilai `LogEventsWithFindings` metrik hanya bertambah satu, karena metrik tersebut menghitung jumlah peristiwa log yang berisi data sensitif, bukan jumlah kejadian data sensitif.

Kebijakan kunci yang diperlukan untuk mengirim temuan audit ke ember yang dilindungi oleh AWS KMS

Anda dapat melindungi data dalam bucket Amazon S3 dengan mengaktifkan Enkripsi Sisi Server dengan Amazon S3-Managed Keys (SSE-S3) atau Enkripsi Sisi Server dengan Kunci KMS (SSE-KMS). Untuk informasi selengkapnya, lihat Melindungi data menggunakan enkripsi sisi server di Panduan Pengguna Amazon S3.

Jika Anda mengirim temuan audit ke bucket yang dilindungi dengan SSE-S3, konfigurasi tambahan tidak diperlukan. Amazon S3 menangani kunci enkripsi.

Jika Anda mengirim temuan audit ke bucket yang dilindungi oleh SSE-KMS, Anda harus memperbarui kebijakan kunci untuk kunci KMS Anda sehingga akun pengiriman log dapat menulis ke bucket S3 Anda. Untuk informasi selengkapnya tentang kebijakan kunci yang diperlukan untuk digunakan dengan SSE-KMS, lihat Amazon S3 di Panduan Pengguna Amazon CloudWatch Logs.

Jenis data yang dapat Anda lindungi

Bagian ini berisi informasi tentang jenis data yang dapat Anda lindungi dalam kebijakan perlindungan data CloudWatch Log. CloudWatch Pengidentifikasi data terkelola log menawarkan tipe data yang

telah dikonfigurasi sebelumnya untuk melindungi data keuangan, informasi kesehatan pribadi (PHI), dan informasi identitas pribadi (PII). Anda juga dapat menggunakan pengidentifikasi data khusus untuk membuat pengidentifikasi data yang disesuaikan dengan kasus penggunaan spesifik Anda.

Daftar Isi

CloudWatch Pengidentifikasi data terkelola log untuk tipe data sensitif

Bagian ini berisi informasi tentang jenis data yang dapat Anda lindungi menggunakan pengidentifikasi data terkelola, dan negara dan wilayah mana yang relevan untuk masing-masing jenis data tersebut.

Untuk beberapa jenis data sensitif, perlindungan data CloudWatch Log memindai kata kunci di dekat data, dan menemukan kecocokan hanya jika menemukan kata kunci itu. Jika kata kunci harus berada

di dekat tipe data tertentu, kata kunci biasanya harus berada dalam 30 karakter (inklusif) dari data tersebut.

Jika kata kunci berisi spasi, perlindungan data CloudWatch Log secara otomatis cocok dengan variasi kata kunci yang kehilangan ruang atau yang berisi garis bawah (_) atau tanda hubung (-) alih-alih spasi. Dalam beberapa kasus, CloudWatch Log juga memperluas atau menyingkat kata kunci untuk mengatasi variasi umum kata kunci.

Tabel berikut mencantumkan jenis informasi kredensyal, perangkat, keuangan, medis, dan kesehatan yang dilindungi (PHI) yang dapat dideteksi oleh CloudWatch Log menggunakan pengidentifikasi data terkelola. Tabel ini merupakan tambahan untuk tipe data tertentu yang mungkin juga memenuhi syarat sebagai informasi pengenal pribadi (PII).

Pengidentifikasi yang didukung yang independen bahasa dan wilayah

Pengidentifikasi	Kategori
`Address`	Pribadi
`AwsSecretKey`	Kredensial
`CreditCardExpiration`	Keuangan
`CreditCardNumber`	Keuangan
`CreditCardSecurityCode`	Keuangan
`EmailAddress`	Pribadi
`IpAddress`	Pribadi
`LatLong`	Pribadi
`Name`	Pribadi
`OpenSshPrivateKey`	Kredensial
`PgpPrivateKey`	Kredensial
`PkcsPrivateKey`	Kredensial

Pengidentifikasi	Kategori
`PuttyPrivateKey`	Kredensial
`VehicleIdentificationNumber`	Pribadi

Pengidentifikasi data yang bergantung pada wilayah harus menyertakan nama pengenal, lalu tanda hubung, dan kemudian kode dua huruf (ISO 3166-1 alpha-2). Misalnya, `DriversLicense-US`.

Pengidentifikasi yang didukung yang harus menyertakan kode negara atau wilayah dua huruf

Pengidentifikasi	Kategori	Negara dan bahasa
BankAccountNumber	Keuangan	DE, ES, FR, GB, ITU
CepCode	Pribadi	BR
Cnpj	Pribadi	BR
CpfCode	Pribadi	BR
DriversLicense	Pribadi	DI, AU, BE, BG, CA, CY, CZ, DE, DK, E, ES, FI, FR, GB, GR, HR, HU, YAITU, ITU, LT, LU, LV, MT, NL, PL, PT, RO, SE, SI, SK, US
DrugEnforcementAgencyNumber	Kondisi	AS
ElectoralRollNumber	Pribadi	GB
HealthInsuranceCardNumber	Kondisi	EU
HealthInsuranceClaimNumber	Kondisi	AS
HealthInsuranceNumber	Kondisi	FR
HealthcareProcedureCode	Kondisi	AS

Pengidentifikasi	Kategori	Negara dan bahasa
IndividualTaxIdentification Number	Pribadi	AS
InseeCode	Pribadi	FR
MedicareBeneficiaryNumber	Kondisi	AS
NationalDrugCode	Kondisi	AS
NationalIdentificationNumber	Pribadi	DE, ES, ITU
NationalInsuranceNumber	Pribadi	GB
NationalProviderId	Kondisi	AS
NhsNumber	Kondisi	GB
NieNumber	Pribadi	ES
NifNumber	Pribadi	ES
PassportNumber	Pribadi	CA, DE, ES, FR, GB, ITU, KAMI
PermanentResidenceNumber	Pribadi	CA
PersonalHealthNumber	Kondisi	CA
PhoneNumber	Pribadi	BR, DE, ES, FR, GB, ITU, KAMI
PostalCode	Pribadi	CA
RgNumber	Pribadi	BR
SocialInsuranceNumber	Pribadi	CA
Ssn	Pribadi	ES, KITA
TaxId	Pribadi	DE, ES, FR, GB

Pengidentifikasi	Kategori	Negara dan bahasa
ZipCode	Pribadi	AS

Kredensial

CloudWatch Perlindungan data log dapat menemukan jenis kredensil berikut.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah
AWS kunci akses rahasia	`AwsSecretKey`	`aws_secret_access_key` , `credentials` , `secret access key`, `secret key`, `set-awscredential`	Semua
Kunci pribadi OpenSSH	`OpenSSHPrivateKey`	Tidak ada	Semua
Kunci pribadi PGP	`PgpPrivateKey`	Tidak ada	Semua
Kunci Pribadi PKCS	`PkcsPrivateKey`	Tidak ada	Semua
Kunci pribadi PuTTY	`PuttyPrivateKey`	Tidak ada	Semua

ARN pengenal data untuk tipe data kredensi

Berikut ini mencantumkan Nama Sumber Daya Amazon (ARN) untuk pengidentifikasi data yang dapat Anda tambahkan ke kebijakan perlindungan data Anda.

ARN pengidentifikasi data kredensi
`arn:aws:dataprotection::aws:data-identifier/AwsSecretKey`

ARN pengidentifikasi data kredensi
`arn:aws:dataprotection::aws:data-identifier/OpenSshPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PgpPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PkcsPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PuttyPrivateKey`

Pengidentifikasi perangkat

CloudWatch Perlindungan data log dapat menemukan jenis pengidentifikasi perangkat berikut.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah
Alamat IP	`IpAddress`	Tidak ada	Semua

ARN pengenal data untuk tipe data perangkat

Berikut ini mencantumkan Nama Sumber Daya Amazon (ARN) untuk pengidentifikasi data yang dapat Anda tambahkan ke kebijakan perlindungan data Anda.

Pengenal data perangkat ARN
`arn:aws:dataprotection::aws:data-identifier/IpAddress`

Informasi keuangan

CloudWatch Perlindungan data log dapat menemukan jenis informasi keuangan berikut.

Jika Anda menetapkan kebijakan perlindungan data, CloudWatch Log akan memindai pengenal data yang Anda tentukan, apa pun geolokasi grup log tersebut berada. Informasi di kolom Negara dan wilayah dalam tabel ini menunjukkan apakah kode negara dua huruf harus ditambahkan ke pengenal data untuk mendeteksi kata kunci yang sesuai untuk negara dan wilayah tersebut.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nomor rekening bank	`BankAccountNumber`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel Kata kunci untuk nomor rekening bank nanti di bagian ini.	Prancis, Jerman, Italia, Spanyol, Inggris	Termasuk International Bank Account Nmbers (IBAN) yang terdiri dari hingga 34 karakter alfanumerik, termasuk elemen seperti kode negara.
Tanggal kedaluwarsa kartu kredit	`CreditCardExpiration`	`exp d`, `exp m`, `exp y`, `expiration` , `expiry`	Semua	
Nomor kartu kredit	`CreditCardNumber`	`account number`, `american express`, `amex`, `bank card`, `card`, `card number`, `card num`, `cc #`, `ccn`, `check card`, `credit`, `credit card#`, `dankort`, `debit`, `debit card`, `diners`	Semua	Deteksi mengharuskan data menjadi urutan 13-19 digit

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
		`club`, `discover`, `electron`, `japanese card bureau`, `jcb`, `mastercard` , `mc`, `pan`, `payment account number`, `payment card number`, `pcn`, `union pay`, `visa`		yang mematuhi rumus pemeriksaan Luhn, dan menggunakan awalan nomor kartu standar untuk salah satu jenis kartu kredit berikut: American Express, Dankort, Diner's Club, Discover, Electron, Japanese Card Bureau (JCB), Mastercard, dan

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
				Visa. UnionPay
Kode verifikasi kartu kredit	`CreditCardSecurityCode`	`card id`, `card identification code`, `card identification number` , `card security code`, `card validation code` , `card validation number` , `card verification data` , `card verification value`, `cvc`, `cvc2`, `cvv`, `cvv2`, `elo verification code`	Semua	

Kata kunci untuk nomor rekening bank

Gunakan kata kunci berikut untuk mendeteksi Nomor Rekening Bank Internasional (IBAN) yang terdiri dari hingga 34 karakter alfanumerik, termasuk elemen seperti kode negara.

Negara	Kata kunci
France	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `code bancaire`, `compte bancaire`, `customer account id`, `customer account number`, `customer bank account id`, `iban`, `numéro de compte`
Germany	`account code`, `account number`, `accountno#` , `accountnumber#` , `bankleitzahl` , `bban`, `customer account id`, `customer account number`, `customer bank account id`, `geheimzahl` , `iban`, `kartennummer` , `kontonummer` , `kreditkartennummer` , `sepa`

Negara	Kata kunci
Italy	account code, account number, accountno# , accountnumber# , bban, codice bancario, conto bancario, customer account id, customer account number, customer bank account id, iban, numero di conto
Spain	account code, account number, accountno# , accountnumber# , bban, código cuenta, código cuenta bancaria, cuenta cliente id, customer account ID, customer account number, customer bank account id, iban, número cuenta bancaria cliente, número cuenta cliente
Britania Raya	account code, account number, accountno# , accountnumber# , bban, customer account ID, customer account number, customer bank account id, iban, sepa
Amerika Serikat	bank account, bank acct, checking account, checking acct, deposit account, deposit acct, savings account, savings acct, chequing account, chequing acct

CloudWatch Log tidak melaporkan kejadian urutan berikut, yang telah disediakan oleh penerbit kartu kredit untuk pengujian publik.

```
122000000000003, 2222405343248877, 2222990905257051, 2223007648726984,
 2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
 36148900647913,
36700102000000, 371449635398431, 378282246310005, 378734493671000, 38520000023237,
 4012888888881881,
4111111111111111, 4222222222222, 4444333322221111, 4462030000000000, 4484070000000000,
 4911830000000,
4917300800000000, 4917610000000000, 4917610000000000003, 5019717010103742,
 5105105105105100,
5111010030175156, 5185540810000019, 5200828282828210, 5204230080000017,
 5204740009900014, 5420923878724339,
5454545454545454, 5455330760000018, 5506900490000436, 5506900490000444,
 5506900510000234, 5506920809243667,
```

```
5506922400634930, 5506927427317625, 5553042241984105, 5555553753048194,
 5555555555554444, 5610591081018250,
6011000990139424, 6011000400000000, 6011111111111117, 630490017740292441,
 630495060000000000,
6331101999990016, 6759649826438453, 6799990100000000019, and 76009244561.
```

ARN pengenal data untuk tipe data keuangan

Berikut ini mencantumkan Nama Sumber Daya Amazon (ARN) untuk pengidentifikasi data yang dapat Anda tambahkan ke kebijakan perlindungan data Anda.

ARN pengidentifikasi data keuangan
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-DE`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-ES`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-FR`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-GB`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-IT`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-US`
`arn:aws:dataprotection::aws:data-identifier/CreditCardExpiration`
`arn:aws:dataprotection::aws:data-identifier/CreditCardNumber`
`arn:aws:dataprotection::aws:data-identifier/CreditCardSecurityC ode`

Informasi kesehatan yang dilindungi (PHI)

CloudWatch Perlindungan data log dapat menemukan jenis informasi kesehatan yang dilindungi (PHI) berikut.

Jika Anda menetapkan kebijakan perlindungan data, CloudWatch Log akan memindai pengenal data yang Anda tentukan, apa pun geolokasi grup log tersebut berada. Informasi di kolom Negara dan wilayah dalam tabel ini menunjukkan apakah kode negara dua huruf harus ditambahkan ke pengenal data untuk mendeteksi kata kunci yang sesuai untuk negara dan wilayah tersebut.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah
Nomor registrasi Badan Penegakan Narkoba (DEA)	DrugEnforcementAgencyNumber	dea number, dea registration	Amerika Serikat
Nomor Kartu Asuransi Kesehatan (EHIC)	HealthInsuranceCardNumber	assicurazione sanitaria numero, carta assicurazione numero, carte d’assurance maladie , carte européenne d'assurance maladie , ceam, ehic, ehic#, finlandehicnumber# , gesundheitskarte , hälsokort , health card, health card number, health insurance card, health insurance number, insurance card number, krankenversicherungskarte , krankenversicherungsnummer , medical account number, numero conto medico, numéro d’assurance maladie , numéro de carte d’assurance , numéro de compte medical, número de cuenta médica, número de seguro de salud, número de tarjeta	Uni Eropa

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah
		de seguro, sairaanhoitokortin , sairausvakuutuskortti , sairausvakuutusnumero , sjukförsäkringnummer, sjukförsäkringskort , suomi ehic-numero , tarjeta de salud, terveyskortti , tessera sanitaria assicurazione numero , versicherungsnummer	
Nomor Klaim Asuransi Kesehatan (HICN)	HealthInsuranceClaimNumber	health insurance claim number, hic no, hic no., hic number, hic#, hicn, hicn#, hicno#	Amerika Serikat
Nomor asuransi atau identifikasi medis	HealthInsuranceNumber	carte d'assuré social, carte vitale, insurance card	France
Kode Sistem Pengkodean Prosedur Umum Pemeliharaan Kesehatan (HCPCS)	HealthcareProcedureCode	current procedural terminology , hcpcs, healthcare common procedure coding system	Amerika Serikat
Nomor Penerima Medicare (MBN)	MedicareBeneficiaryNumber	mbi, medicare beneficiary	Amerika Serikat

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah
Kode Obat Nasional (NDC)	`NationalDrugCode`	`national drug code`, `ndc`	Amerika Serikat
Pengidentifikasi Penyedia Nasional (NPI)	`NationalProviderId`	`hipaa`, `n.p.i.`, `national provider`, `npi`	Amerika Serikat
Nomor Layanan Kesehatan Nasional (NHS)	`NhsNumber`	`national health service`, `NHS`	Inggris Raya
Nomor Kesehatan Pribadi	`PersonalHealthNumber`	`canada healthcare number`, `msp number`, `care number`, `phn`, `soins de santé`	Kanada

ARN pengidentifikasi data untuk tipe data informasi kesehatan yang dilindungi (PHI)

Berikut ini mencantumkan pengenal data Nama Sumber Daya Amazon (ARN) yang dapat digunakan dalam kebijakan perlindungan data informasi kesehatan yang dilindungi (PHI).

ARN pengidentifikasi data PHI
`arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US`
`arn:aws:dataprotection::aws:data-identifier/HealthcareProcedureCode-US`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceCardNumber-EU`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceClaimNumber-US`

ARN pengidentifikasi data PHI
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceNumber-FR
arn:aws:dataprotection::aws:data-identifier/MedicareBeneficiaryNumber-US
arn:aws:dataprotection::aws:data-identifier/NationalDrugCode-US
arn:aws:dataprotection::aws:data-identifier/NationalInsuranceNumber-GB
arn:aws:dataprotection::aws:data-identifier/NationalProviderId-US
arn:aws:dataprotection::aws:data-identifier/NhsNumber-GB
arn:aws:dataprotection::aws:data-identifier/PersonalHealthNumber-CA

Informasi Identifikasi Pribadi (PII)

CloudWatch Perlindungan data log dapat menemukan jenis informasi identitas pribadi (PII) berikut.

Jika Anda menetapkan kebijakan perlindungan data, CloudWatch Log akan memindai pengenal data yang Anda tentukan, apa pun geolokasi grup log tersebut berada. Informasi di kolom Negara dan wilayah dalam tabel ini menunjukkan apakah kode negara dua huruf harus ditambahkan ke pengenal data untuk mendeteksi kata kunci yang sesuai untuk negara dan wilayah tersebut.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Tanggal lahir	DateOfBirth	dob, date of birth, birthdate , birth date, birthday, b-day, bday	Setiap	Support mencakup sebagian besar format tanggal,

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
				seperti semua digit dan kombinasi digit dan nama bulan. Komponen tanggal dapat dipisahkan oleh spasi, garis miring (/), atau tanda hubung (-).
Kode Pos Endereçamento (CEP)	`CepCode`	`cep`, `código de endereçamento postal`, `codigo de endereçamento postal`	Brazil	
Kadastro Nacional da Pessoa Jurídica (CNPJ)	`Cnpj`	`cadastro nacional da pessoa jurídica`, `cadastro nacional da pessoa juridica`, `cnpj`	Brazil	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Kadaster Pessoas Físicas (CPF)	`CpfCode`	`Cadastro de pessoas fisicas`, `cadastro de pessoas físicas`, `cadastro de pessoa física`, `cadastro de pessoa fisica`, `cpf`	Brazil	
Nomor identifikasi lisensi	`DriversLicense`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel nomor identifikasi SIM nanti di bagian ini.	Banyak negara. Untuk detailnya, lihat tabel nomor identifikasi SIM.	
Nomor Roll Pemilu	`ElectoralRollNumber`	`electoral #`, `electoral number`, `electoral roll #`, `electoral roll no.`, `electoral roll number`, `electoral rollno`	Britania Raya	
Identifikasi wajib pajak individu	`IndividualTaxIdentificationNumber`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel nomor identifikasi wajib pajak perorangan nanti di bagian ini.	Brasil, Prancis, Jerman, Spanyol, Inggris	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Institut Nasional untuk Statistik dan Studi Ekonomi (INSEE)	`InseeCode`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel Kata kunci untuk nomor identifikasi nasional nanti di bagian ini.	France	
Nomor Identifikasi Nasional	`NationalIdentificationNumber`	Ya. Untuk detailnya, lihat tabel Kata kunci untuk nomor identifikasi nasional nanti di bagian ini.	Jerman, Italia, Spanyol	Ini termasuk pengidentifikasi Documento Nacional de Identidad (DNI) (Spanyol), kode fiscale Codice (Italia), dan nomor Kartu Identitas Nasional (Jerman).

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nomor Asuransi Nasional (NINO)	`NationalInsuranceNumber`	`insurance no.`, `insurance number`, `insurance#` , `national insurance number`, `nationalinsurance#` , `nationalinsurancenumber` , `nin`, `nino`	Britania Raya	–
Nama Identidad Extranjero (NIE)	`NieNumber`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel nomor identifikasi wajib pajak perorangan nanti di bagian ini.	Spain	
Nomor Identifikasi Fiskal (NIF)	`NifNumber`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel nomor identifikasi wajib pajak perorangan nanti di bagian ini.	Spain	
Nomor paspor	`PassportNumber`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel Kata kunci untuk nomor paspor nanti di bagian ini.	Kanada, Prancis, Jerman, Italia, Spanyol, Inggris, Amerika Serikat	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nomor tempat tinggal permanen	`Permanent Residence Number`	`carte résident permanent , numéro carte résident permanent , numéro résident permanent , permanent resident card, permanent resident card number, permanent resident no, permanent resident no., permanent resident number, pr no, pr no., pr non, pr number, résident permanent no., résident permanent non`	Kanada	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nomor telepon	PhoneNumber	Brasil: kata kunci juga meliputi: cel,celular,fone,,móvel,, residencial ,numero residencial , telefone Lainnya:cell,contact,fax, number,mobile,phone,pho number,tel,telephone , telephone number	Brasil, Kanada, Prancis, Jerman, Italia, Spanyol, Inggris, Amerika Serikat	Ini termasuk nomor bebas pulsa di Amerika Serikat dan nomor faks. Jika kata kunci berada di dekat data, nomor tersebut tidak harus menyertakan kode negara. Jika kata kunci tidak dekat dengan

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
				data, nomor tersebut harus menyertakan kode negara.
Kode Pos	`PostalCode`	Tidak ada	Kanada	
Registrasi Geral (RG)	`RgNumber`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel nomor identifikasi wajib pajak perorangan nanti di bagian ini.	Brazil	
Nomor Pokok Wajib Pajak (SIN)	`SocialInsuranceNumber`	`canadian id`, `numéro d'assurance sociale`, `social insurance number`, `sin`	Kanada	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nomor Jaminan Sosial (SSN)	`Ssn`	Spanyol —`número de la seguridad social`,`social security no.`,`social security no.` `número de la seguridad social`,`social security number`,`socialsecurityno#` ,`ssn`, `ssn#` Amerika Serikat -`social security`,`ss#`, `ssn`	Spanyol, Amerika Serikat	
Nomor identifikasi wajib pajak atau referensi	`TaxId`	Ya. Kata kunci yang berbeda berlaku untuk negara yang berbeda. Untuk detailnya, lihat tabel nomor identifikasi wajib pajak perorangan nanti di bagian ini. .	Prancis, Jerman, Spanyol, Inggris	Ini termasuk TIN (Prancis); Steueridentifikationsnummer (Jerman); CIF (Spanyol); dan TRN, UTR (Inggris). .

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Kode Pos	`ZipCode`	`zip code`, `zip+4`	Amerika Serikat	Kode pos Amerika Serikat.
Alamat surat-menyurat	`Address`	Tidak ada	Australia, Kanada, Prancis, Jerman, Italia, Spanyol, Inggris, Amerika Serikat	Meskipun kata kunci tidak diperlukan, deteksi memerlukan alamat untuk menyertakan nama kota atau tempat dan kode pos atau kode pos.
Alamat surat elektronik	`EmailAddress`	Tidak ada	Setiap	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Koordinat Global Positioning System (GPS)	`LatLong`	`coordinate` , `coordinates` , `lat long`, `latitude longitude` , `location`, `position`	Setiap	CloudWatch Log dapat mendeteksi koordinat GPS jika koordinat lintang dan bujur disimpan sebagai pasangan dan mereka dalam format Derajat Desimal (DD), misalnya, 41.948614 , -87.655311. Support tidak menyertakan koordinat

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
				dalam format Degrees Decimal Minutes (DDM), misalnya format 41° 56.9168'N 87° 39.3187'W, atau Derajat, Menit, Detik (DMS), misalnya 41° 56'55.0104 “N 87° 39'19.1196"W.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nama lengkap	`Name`	Tidak ada	Setiap	CloudWatch Log hanya dapat mendeteksi nama lengkap. Dukungan terbatas pada set karakter Latin.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nomor Identifikasi Kendaraan (VIN)	`VehicleIdentificationNumber`	`Fahrgestellnummer` , `niv`, `numarul de identificare` , `numarul seriei de sasiu`, `serie sasiu`, `numer VIN`, `Número de Identificação do Veículo`, `Número de Identificación de Automóviles` , `numéro d'identification du véhicule`, `vehicle identification number`, `vin`, `VIN numeris`	Setiap	CloudWatch Log dapat mendeteksi VIN yang terdiri dari urutan 17 karakter dan mematuhi standar ISO 3779 dan 3780. Standar ini dirancang untuk penggunaan di seluruh dunia.

Kata kunci untuk nomor identifikasi surat izin mengemudi

Untuk mendeteksi berbagai jenis nomor identifikasi SIM, CloudWatch Log membutuhkan kata kunci untuk berada di dekat nomor. Tabel berikut mencantumkan kata kunci yang dikenali CloudWatch Log untuk negara dan wilayah tertentu.

Negara atau wilayah	Kata kunci
Australia	dl# dl:, dl :, dlno# driver licence, driver license, driver permit, drivers lic., drivers licence, driver's licence, drivers license, driver's license, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Austria	führerschein, fuhrerschein, führerschein republik österreich, fuhrerschein republik osterreich
Belgium	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer, numéro permis conduire, permis de conduire, rijbewijs, rijbewijsnummer
Bulgaria	превозно средство, свидетелство за управление на моторно, свидетелство за управление на мпс, сумпс, шофьорска книжка
Canada	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit, permis de conduire
Croatia	vozačka dozvola
Cyprus	άδεια οδήγησης

Negara atau wilayah	Kata kunci
Czech Republic	číslo licence, číslo licence řidiče, číslo řidičskéh o průkazu, ovladače lic., povolení k jízdě, povolení řidiče, řidiči povolení, řidičský prúkaz, řidičský průkaz
Denmark	kørekort, kørekortnummer
Estonia	juhi litsentsi number, juhiloa number, juhiluba, juhiluba number
Finland	ajokortin numero, ajokortti, förare lic., körkort, körkort nummer, kuljettaja lic., permis de conduire
France	permis de conduire
Germany	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer
Greece	δεια οδήγησης, adeia odigisis
Hungary	illesztőprogramok lic, jogosítvány, jogsi, licencszám, vezető engedély, vezetői engedély
Ireland	ceadúnas tiomána
Italy	patente di guida, patente di guida numero, patente guida, patente guida numero
Latvia	autovadītāja apliecība, licences numurs, vadītāja apliecība, vadītāja apliecības numurs, vadītāja atļauja, vadītāja licences numurs, vadītāji lic.
Lithuania	vairuotojo pažymėjimas

Negara atau wilayah	Kata kunci
Luxembourg	fahrerlaubnis, führerschäin
Malta	liċenzja tas-sewqan
Netherlands	permis de conduire, rijbewijs, rijbewijsnummer
Poland	numer licencyjny, prawo jazdy, zezwolenie na prowadzenie
Portugal	carta de condução, carteira de habilitação, carteira de motorist, carteira habilitação, carteira motorist, licença condução, licença de condução, número de licença, número licença, permissão condução, permissão de condução
Romania	numărul permisului de conducere, permis de conducere
Slovakia	číslo licencie, číslo vodičského preukazu, ovládače lic., povolenia vodičov, povolenie jazdu, povolenie na jazdu, povolenie vodiča, vodičský preukaz
Slovenia	vozniško dovoljenje
Spain	carnet conducer, el carnet de conducer, licencia conducer, licencia de manejo, número carnet conducer, número de carnet de conducer, número de permiso conducer, número de permiso de conducer, número licencia conducer, número permiso conducer, permiso conducción, permiso conducer, permiso de conducción
Sweden	ajokortin numero, dlno# ajokortti, drivere lic., förare lic., körkort, körkort nummer, körkortsn ummer, kuljettajat lic.

Negara atau wilayah	Kata kunci
Britania Raya	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Amerika Serikat	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit

Kata kunci untuk nomor induk kependudukan

Untuk mendeteksi berbagai jenis nomor identifikasi nasional, CloudWatch Log membutuhkan kata kunci untuk berada di dekat nomor. Hal ini termasuk pengenal Documento Nacional de Identidad (DNI) (Spain), kode French National Institute for Statistics and Economic Studies (INSEE), nomor German National Identity Card, dan nomor Registro Geral (RG) (Brazil).

Tabel berikut mencantumkan kata kunci yang dikenali CloudWatch Log untuk negara dan wilayah tertentu.

Negara atau wilayah	Kata kunci
Brazil	registro geral, rg
France	assurance sociale, carte nationale d’identit é, cni, code sécurité sociale, French social security number, fssn#, insee, insurance

Negara atau wilayah	Kata kunci
	number, national id number, nationalid#, numéro d'assurance, sécurité sociale, sécurité sociale non., sécurité sociale numéro, social, social security, social security number, socialsecuritynumber, ss#, ssn, ssn#
Germany	ausweisnummer, id number, identification number, identity number, insurance number, personal id, personalausweis
Italy	codice fiscal, dati anagrafici, ehic, health card, health insurance card, p. iva, partita i.v.a., personal data, tax code, tessera sanitaria
Spain	dni, dni#, dninúmero#, documento nacional de identidad, identidad único, identidadúnico#, insurance number, national identification number, national identity, nationalid#, nationali dno#, número nacional identidad, personal identification number, personal identity no, unique identity number, uniqueid#

Kata kunci untuk nomor paspor

Untuk mendeteksi berbagai jenis nomor paspor, CloudWatch Log membutuhkan kata kunci untuk berada di dekat nomor. Tabel berikut mencantumkan kata kunci yang dikenali CloudWatch Log untuk negara dan wilayah tertentu.

Negara atau wilayah	Kata kunci
Canada	passeport, passeport#, passport, passport#, passportno, passportno#
France	numéro de passeport, passeport, passeport #, passeport #, passeportn °, passeport n °, passeportNon, passeport non

Negara atau wilayah	Kata kunci
Germany	ausstellungsdatum, ausstellungsort, geburtsdatum, passport, passports, reisepass, reisepass–nr, reisepassnummer
Italy	italian passport number, numéro passeport , numéro passeport italien, passaporto, passaporto italiana, passaporto numero, passport number, repubblica italiana passaporto
Spain	españa pasaporte, libreta pasaporte, número pasaporte, pasaporte, passport, passport book, passport no, passport number, spain passport
Britania Raya	passeport #, passeport n °, passeportNon, passeport non, passeportn °, passport #, passport no, passport number, passport#, passportid
Amerika Serikat	passport, travel document

Kata kunci untuk nomor pokok wajib pajak

Untuk mendeteksi berbagai jenis identifikasi wajib pajak dan nomor referensi, CloudWatch Log membutuhkan kata kunci untuk berada di dekat angka-angka tersebut. Tabel berikut mencantumkan kata kunci yang dikenali CloudWatch Log untuk negara dan wilayah tertentu.

Negara atau wilayah	Kata kunci
Brazil	cadastro de pessoa física, cadastro de pessoa fisica, cadastro de pessoas físicas, cadastro de pessoas fisicas, cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj, cpf

Negara atau wilayah	Kata kunci
France	numéro d'identification fiscale, tax id, tax identification number, tax number, tin, tin#
Germany	identifikationsnummer, steuer id, steueride ntifikationsnummer, steuernummer, tax id, tax identification number, tax number
Spain	cif, cif número, cifnúmero#, nie, nif, número de contribuyente, número de identidad de extranjero, número de identificación fiscal, número de impuesto corporativo, personal tax number, tax id, tax identification number, tax number, tin, tin#
Britania Raya	paye, tax id, tax id no., tax id number, tax identification, tax identification#, tax no., tax number, tax reference, tax#, taxid#, temporary reference number, tin, trn, unique tax reference, unique taxpayer reference, utr
Amerika Serikat	nomor identifikasi wajib pajak individu, itin, i.t.i.n.

ARN pengidentifikasi data untuk informasi identitas pribadi (PII)

Tabel berikut mencantumkan Nama Sumber Daya Amazon (ARN) untuk pengidentifikasi data informasi identitas pribadi (PII) yang dapat Anda tambahkan ke kebijakan perlindungan data Anda.

ARN pengidentifikasi data PII
`arn:aws:dataprotection::aws:data-identifier/Address`
`arn:aws:dataprotection::aws:data-identifier/CepCode-BR`
`arn:aws:dataprotection::aws:data-identifier/Cnpj-BR`

ARN pengidentifikasi data PII
arn:aws:dataprotection::aws:data-identifier/CpfCode-BR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-AT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-AU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-BE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-BG
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CA
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CY
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CZ
arn:aws:dataprotection::aws:data-identifier/DriversLicense-DE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-DK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-EE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-ES
arn:aws:dataprotection::aws:data-identifier/DriversLicense-FI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-FR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GB
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IT

ARN pengidentifikasi data PII
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LV
arn:aws:dataprotection::aws:data-identifier/DriversLicense-MT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-NL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-RO
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-US
arn:aws:dataprotection::aws:data-identifier/ElectoralRollNumber-GB
arn:aws:dataprotection::aws:data-identifier/EmailAddress
arn:aws:dataprotection::aws:data-identifier/IndividualTaxIdentificationNumber-US
arn:aws:dataprotection::aws:data-identifier/InseeCode-FR
arn:aws:dataprotection::aws:data-identifier/LatLong
arn:aws:dataprotection::aws:data-identifier/Name

ARN pengidentifikasi data PII
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-DE
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-ES
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-IT
arn:aws:dataprotection::aws:data-identifier/NieNumber-ES
arn:aws:dataprotection::aws:data-identifier/NifNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-CA
arn:aws:dataprotection::aws:data-identifier/PassportNumber-DE
arn:aws:dataprotection::aws:data-identifier/PassportNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-FR
arn:aws:dataprotection::aws:data-identifier/PassportNumber-GB
arn:aws:dataprotection::aws:data-identifier/PassportNumber-IT
arn:aws:dataprotection::aws:data-identifier/PassportNumber-US
arn:aws:dataprotection::aws:data-identifier/PermanentResidenceNumber-CA
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-BR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-DE
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-ES
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-FR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-GB

ARN pengidentifikasi data PII
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-IT
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-US
arn:aws:dataprotection::aws:data-identifier/PostalCode-CA
arn:aws:dataprotection::aws:data-identifier/RgNumber-BR
arn:aws:dataprotection::aws:data-identifier/SocialInsuranceNumber-CA
arn:aws:dataprotection::aws:data-identifier/Ssn-ES
arn:aws:dataprotection::aws:data-identifier/Ssn-US
arn:aws:dataprotection::aws:data-identifier/TaxId-DE
arn:aws:dataprotection::aws:data-identifier/TaxId-ES
arn:aws:dataprotection::aws:data-identifier/TaxId-FR
arn:aws:dataprotection::aws:data-identifier/TaxId-GB
arn:aws:dataprotection::aws:data-identifier/VehicleIdentificationNumber
arn:aws:dataprotection::aws:data-identifier/ZipCode-US

Pengidentifikasi data khusus

Topik

- Apa itu pengidentifikasi data khusus?
- Kendala pengenal data kustom
- Menggunakan pengidentifikasi data khusus di konsol
- Menggunakan pengidentifikasi data khusus dalam kebijakan perlindungan data Anda

Apa itu pengidentifikasi data khusus?

Pengidentifikasi data kustom (CDI) memungkinkan Anda menentukan ekspresi reguler kustom Anda sendiri yang dapat digunakan dalam kebijakan perlindungan data Anda. Dengan menggunakan pengidentifikasi data khusus, Anda dapat menargetkan kasus penggunaan informasi identitas pribadi (PII) khusus bisnis yang tidak dapat diberikan oleh pengidentifikasi data terkelola. Misalnya, Anda dapat menggunakan pengenal data khusus untuk mencari ID karyawan khusus perusahaan. Pengidentifikasi data khusus dapat digunakan bersama dengan pengidentifikasi data terkelola.

Kendala pengenal data kustom

CloudWatch Log pengidentifikasi data kustom memiliki batasan berikut:

- Maksimal 10 pengidentifikasi data kustom didukung untuk setiap kebijakan perlindungan data.
- Nama pengidentifikasi data kustom memiliki panjang maksimum 128 karakter. Karakter berikut didukung:
  - Alfanumerik: (A-za-Z0-9)
  - Simbol: ('_' | '-')
- RegExmemiliki panjang maksimum 200 karakter. Karakter berikut didukung:
  - Alfanumerik: (A-za-Z0-9)
  - Simbol: ('_' | '#' | '=' | '@' |'/' | ';' | ',' | '-' | '')
  - RegExkarakter yang dipesan: ('^' | '$' | '?' | '[' | ']' | '{' | '}' | '|'\ '|' * '|' + '|'. ')
- Pengidentifikasi data kustom tidak dapat berbagi nama yang sama dengan pengenal data terkelola.
- Pengidentifikasi data khusus dapat ditentukan dalam kebijakan perlindungan data tingkat akun atau dalam kebijakan perlindungan data tingkat grup log. Mirip dengan pengidentifikasi data terkelola, pengidentifikasi data kustom yang ditentukan dalam kebijakan tingkat akun bekerja dalam kombinasi dengan pengidentifikasi data kustom yang ditentukan dalam kebijakan tingkat grup log.

Menggunakan pengidentifikasi data khusus di konsol

Saat Anda menggunakan CloudWatch konsol untuk membuat atau mengedit kebijakan perlindungan data, untuk menentukan pengenal data kustom, Anda cukup memasukkan nama dan ekspresi reguler untuk pengenal data. Misalnya, Anda mungkin memasukkan **`Employee_ID`** nama dan **`EmployeeID-\d{9}`** sebagai ekspresi reguler. Ekspresi reguler ini akan mendeteksi dan menutupi peristiwa log dengan sembilan angka setelahnya`EmployeeID-`. Misalnya, `EmployeeID-123456789`

Menggunakan pengidentifikasi data khusus dalam kebijakan perlindungan data Anda

Jika Anda menggunakan AWS API AWS CLI atau untuk menentukan pengenal data kustom, Anda harus menyertakan nama pengenal data dan ekspresi reguler dalam kebijakan JSON yang digunakan untuk menentukan kebijakan perlindungan data. Kebijakan perlindungan data berikut mendeteksi dan menutupi peristiwa log yang membawa ID karyawan khusus perusahaan.

1. Buat `Configuration` blok dalam kebijakan perlindungan data Anda.
2. Masukkan a `Name` untuk pengenal data kustom Anda. Misalnya, **`EmployeeId`**.
3. Masukkan a `Regex` untuk pengenal data kustom Anda. Misalnya, **`EmployeeID-\d{9}`**. Ekspresi reguler ini akan cocok dengan peristiwa log `EmployeeID-` yang berisi sembilan digit `EmployeeID-` setelahnya. Misalnya, `EmployeeID-123456789`
4. Lihat pengenal data kustom berikut dalam pernyataan kebijakan.

```
{
    "Name": "example_data_protection_policy",
    "Description": "Example data protection policy with custom data identifiers",
    "Version": "2021-06-01",
    "Configuration": {
      "CustomDataIdentifier": [
        {"Name": "EmployeeId", "Regex": "EmployeeId-\\d{9}"}
      ]
    },
    "Statement": [
        {
            "Sid": "audit-policy",
            "DataIdentifier": [
                "EmployeeId"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
```

```
            "DataIdentifier": [
            "EmployeeId"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {
                    }
                }
            }
        }
    ]
}
```

5. (Opsional) Lanjutkan untuk menambahkan pengidentifikasi data kustom tambahan ke `Configuration` blok sesuai kebutuhan. Kebijakan perlindungan data saat ini mendukung maksimal 10 pengidentifikasi data kustom.

Membuat metrik dari peristiwa log menggunakan filter

Anda dapat mencari dan memfilter data log yang masuk ke CloudWatch Log dengan membuat satu atau beberapa filter metrik. Filter metrik menentukan istilah dan pola yang harus dicari dalam data log saat dikirim ke CloudWatch Log. CloudWatch Log menggunakan filter metrik ini untuk mengubah data log menjadi CloudWatch metrik numerik yang dapat Anda buat grafik atau nyalakan alarm.

Saat membuat metrik dari filter log, Anda juga dapat memilih untuk menetapkan dimensi dan unit ke metrik. Jika Anda menentukan unit, pastikan untuk menentukan yang benar saat Anda membuat filter. Mengubah unit untuk filter nanti tidak akan berpengaruh.

> **Note**
>
> Filter metrik hanya didukung untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

Anda dapat menggunakan semua jenis CloudWatch statistik, termasuk statistik persentil, saat melihat metrik ini atau mengatur alarm.

> **Note**
>
> Statistik persentil hanya didukung untuk metrik jika tidak ada nilai metrik yang negatif. Jika Anda mengatur filter metrik sehingga dapat melaporkan angka negatif, statistik persentil tidak akan tersedia untuk metrik tersebut saat ada angka negatif sebagai nilai. Untuk informasi selengkapnya, lihat Persentil.

Filter tidak memfilter data secara retroaktif. Filter hanya memublikasikan titik data metrik untuk kejadian yang terjadi setelah filter dibuat. Hasil yang difilter mengembalikan 50 baris pertama, yang tidak akan ditampilkan jika stempel waktu hasil yang difilter lebih awal daripada waktu pembuatan metrik.

Daftar Isi

- Daftar filter metrik
- Menghapus filter metrik

Konsep

Setiap filter metrik terdiri dari elemen kunci berikut:

nilai default

Nilai yang dilaporkan ke filter metrik selama periode ketika log dicerna tetapi tidak ada log yang cocok ditemukan. Dengan menyetel ini ke 0, Anda memastikan bahwa data dilaporkan selama setiap periode tersebut, mencegah metrik “jerawatan” dengan periode tanpa data yang cocok. Jika tidak ada log yang tertelan selama periode satu menit, maka tidak ada nilai yang dilaporkan.

Jika Anda menetapkan dimensi ke metrik yang dibuat oleh filter metrik, Anda tidak dapat menetapkan nilai default untuk metrik tersebut.

dimensi

Dimensi adalah pasangan kunci-nilai yang menentukan metrik lebih lanjut. Anda dapat menetapkan dimensi ke metrik yang dibuat dari filter metrik. Karena dimensi adalah bagian dari pengidentifikasi unik untuk metrik, setiap kali pasangan nama/nilai unik diekstraksi dari log, Anda membuat variasi baru dari metrik tersebut.

pola filter

Deskripsi simbolis tentang bagaimana CloudWatch Log harus menafsirkan data di setiap peristiwa log. Sebagai contoh, entri log mungkin berisi stempel waktu, alamat IP, string, dan sebagainya. Anda menggunakan pola untuk menentukan apa yang harus dicari dalam berkas log.

nama metrik

Nama CloudWatch metrik tempat informasi log yang dipantau harus dipublikasikan. Misalnya, Anda dapat mempublikasikan ke metrik yang disebut ErrorCount.

namespace metrik

Namespace tujuan metrik baru CloudWatch .

nilai metrik

Nilai numerik untuk dipublikasikan ke metrik setiap kali log yang cocok ditemukan. Sebagai contoh, jika Anda menghitung kemunculan istilah tertentu, seperti "Error", nilainya adalah "1"

untuk setiap kejadian. Jika Anda menghitung byte yang ditransfer, Anda dapat menambahkannya berdasarkan jumlah aktual byte yang ditemukan dalam log acara.

Filter sintaks pola untuk filter metrik

> **Note**
>
> Bagaimana filter metrik berbeda kueri Wawasan CloudWatch Log
> Filter metrik berbeda dari kueri Wawasan CloudWatch Log karena nilai numerik tertentu ditambahkan ke filter metrik setiap kali log yang cocok ditemukan. Untuk informasi selengkapnya, lihat Mengkonfigurasi nilai metrik untuk filter metrik.
> Untuk informasi tentang cara menanyakan grup log Anda dengan bahasa kueri Amazon CloudWatch Logs Insights, lihatCloudWatch Sintaks kueri Log Insights.
> Contoh pola filter generik
> Untuk informasi selengkapnya tentang sintaks pola filter generik yang berlaku untuk filter metrik serta filter langganan dan peristiwa log filter, lihat Filter sintaks pola untuk filter metrik, filter langganan, dan peristiwa log filter, yang mencakup contoh berikut:
>
> - Sintaks ekspresi reguler (regex) yang didukung
> - Istilah yang cocok dalam peristiwa log tidak terstruktur
> - Ketentuan yang cocok dalam peristiwa log JSON
> - Istilah pencocokan dalam peristiwa log yang dibatasi ruang

Filter metrik memungkinkan Anda untuk mencari dan memfilter data log yang masuk ke CloudWatch Log, mengekstrak pengamatan metrik dari data log yang difilter, dan mengubah titik data menjadi metrik CloudWatch Log. Anda menentukan istilah dan pola yang harus dicari dalam data log saat dikirim ke CloudWatch Log. Filter metrik ditetapkan untuk grup log, dan semua filter yang ditetapkan ke grup log diterapkan ke pengaliran log mereka.

Ketika filter metrik cocok dengan istilah, itu menambah jumlah metrik dengan nilai numerik tertentu. Misalnya, Anda dapat membuat filter metrik yang menghitung berapa kali kata ERROR terjadi dalam peristiwa log Anda.

Anda dapat menetapkan satuan ukuran dan dimensi ke metrik. Misalnya, jika Anda membuat filter metrik yang menghitung berapa kali kata ERROR terjadi dalam peristiwa log Anda, Anda dapat

menentukan dimensi yang dipanggil `ErrorCode` untuk menunjukkan jumlah total peristiwa log yang berisi kata ERROR dan memfilter data berdasarkan kode kesalahan yang dilaporkan.

> **Tip**
>
> Saat Anda menetapkan satuan ukuran ke metrik, pastikan untuk menentukan yang benar. Jika Anda mengubah unit nanti, perubahan Anda mungkin tidak berlaku. Untuk daftar lengkap unit yang CloudWatch mendukung, lihat MetricDatumdi Referensi Amazon CloudWatch API.

Topik

Mengkonfigurasi nilai metrik untuk filter metrik

Saat Anda membuat filter metrik, Anda menentukan pola filter dan menentukan nilai metrik dan nilai default Anda. Anda dapat mengatur nilai metrik ke angka, pengidentifikasi bernama, atau pengidentifikasi numerik. Jika Anda tidak menentukan nilai default, tidak CloudWatch akan melaporkan data saat filter metrik Anda tidak menemukan kecocokan. Kami menyarankan Anda menentukan nilai default, bahkan jika nilainya 0. Menyetel nilai default membantu CloudWatch melaporkan data dengan lebih akurat dan CloudWatch mencegah agregasi metrik jerawatan. CloudWatch agregat dan melaporkan nilai metrik setiap menit.

Ketika filter metrik Anda menemukan kecocokan dalam peristiwa log Anda, itu menambah jumlah metrik Anda dengan nilai metrik Anda. Jika filter metrik Anda tidak menemukan kecocokan, CloudWatch laporkan nilai default metrik. Misalnya, grup log Anda menerbitkan dua catatan setiap menit, nilai metriknya adalah 1, dan nilai defaultnya adalah 0. Jika filter metrik Anda menemukan kecocokan di kedua catatan log dalam menit pertama, nilai metrik untuk menit itu adalah 2. Jika filter metrik Anda tidak menemukan kecocokan di kedua rekaman selama menit kedua, nilai default untuk menit itu adalah 0. Jika Anda menetapkan dimensi ke metrik yang dihasilkan oleh filter metrik, Anda tidak dapat menentukan nilai default untuk metrik tersebut.

Anda juga dapat mengatur filter metrik untuk menambah metrik dengan nilai yang diekstrak dari peristiwa log, bukan nilai statis. Untuk informasi selengkapnya, lihat Menggunakan nilai dalam peristiwa log untuk menambah nilai metrik.

Menerbitkan dimensi dengan metrik dari nilai di JSON atau peristiwa log yang dibatasi ruang

Anda dapat menggunakan CloudWatch konsol atau AWS CLI untuk membuat filter metrik yang mempublikasikan dimensi dengan metrik yang dihasilkan oleh JSON dan peristiwa log yang dibatasi ruang. Dimensi adalah pasangan nilai nama/nilai dan hanya tersedia untuk JSON dan pola filter yang dibatasi ruang. Anda dapat membuat filter metrik JSON dan spasi-terbatas hingga tiga dimensi. Untuk informasi selengkapnya tentang dimensi dan informasi tentang cara menetapkan dimensi ke metrik, lihat bagian berikut:

- Dimensi dalam panduan CloudWatch Pengguna Amazon
- Contoh: Ekstrak bidang dari log Apache dan tetapkan dimensi di Panduan Pengguna Amazon CloudWatch Logs

> **Important**
>
> Dimensi berisi nilai yang mengumpulkan biaya yang sama dengan metrik kustom.
> Untuk mencegah muatan tak terduga, jangan tentukan bidang kardinalitas tinggi, seperti `IPAddress` atau`requestID`, sebagai dimensi.
> Jika Anda mengekstrak metrik dari peristiwa log, Anda dikenakan biaya untuk metrik khusus. Untuk mencegah Anda mengumpulkan muatan tinggi yang tidak disengaja, Amazon mungkin menonaktifkan filter metrik Anda jika menghasilkan 1000 pasangan nama/nilai yang berbeda untuk dimensi tertentu selama jangka waktu tertentu.
> Anda dapat membuat alarm penagihan yang memberi tahu Anda tentang perkiraan biaya Anda. Untuk informasi selengkapnya, lihat Membuat alarm penagihan untuk memantau perkiraan AWS tagihan Anda.

Memublikasikan dimensi dengan metrik dari log acara JSON

Contoh berikut berisi cuplikan kode yang menjelaskan cara menentukan dimensi dalam filter metrik JSON.

Example: JSON log event

```
{
  "eventType": "UpdateTrail",
```

```
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {"name": "a",
         "id": 1
       },
       {"name": "b",
         "id": 2
       }
  ]

}
```

> **Note**
>
> Jika Anda menguji filter metrik contoh dengan contoh peristiwa log JSON, Anda harus memasukkan contoh log JSON pada satu baris.

Example: Metric filter

Filter metrik menambah metrik setiap kali peristiwa log JSON berisi properti `eventType` dan. `"sourceIPAddress"`

```
{ $.eventType = "*" && $.sourceIPAddress != 123.123.* }
```

Saat Anda membuat filter metrik JSON, Anda dapat menentukan properti apa pun di filter metrik sebagai dimensi. Misalnya, untuk mengatur `eventType` sebagai dimensi, gunakan yang berikut ini:

```
"eventType" : $.eventType
```

Contoh metrik berisi dimensi yang diberi nama`"eventType"`, dan nilai dimensi dalam peristiwa log contoh adalah`"UpdateTrail"`.

Memublikasikan dimensi dengan metrik dari log acara yang dipisahkan dengan spasi

Contoh berikut berisi cuplikan kode yang menjelaskan cara menentukan dimensi dalam filter metrik yang dibatasi ruang.

Example: Space-delimited log event

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

Example: Metric filter

```
[ip, server, username, timestamp, request, status_code, bytes > 1000]
```

Filter metrik menambah metrik ketika peristiwa log yang dibatasi spasi menyertakan salah satu bidang yang ditentukan dalam filter. Misalnya, filter metrik menemukan bidang dan nilai berikut dalam contoh peristiwa log yang dibatasi ruang.

```
{
   "$bytes": "1534",
   "$status_code": "404",

   "$request": "GET /index.html HTTP/1.0",
   "$timestamp": "10/Oct/2000:13:25:15 -0700",
   "$username": "frank",
   "$server": "Prod",
   "$ip": "127.0.0.1"
}
```

Saat Anda membuat filter metrik yang dibatasi spasi, Anda dapat menentukan salah satu bidang dalam filter metrik sebagai dimensi. Misalnya, untuk mengatur `server` sebagai dimensi, gunakan yang berikut ini:

```
"server" : $server
```

Contoh filter metrik memiliki dimensi yang diberi nama`server`, dan nilai dimensi dalam peristiwa log contoh adalah`"Prod"`.

Example: Match terms with AND (&&) and OR (||)

Anda dapat menggunakan operator logika AND (“&&”) dan OR (“||”) untuk membuat filter metrik yang dibatasi spasi yang berisi kondisi. Filter metrik berikut mengembalikan peristiwa log di mana kata pertama dalam peristiwa adalah ERROR atau superstring dari WARN.

```
[w1=ERROR || w1=%WARN%, w2]
```

Menggunakan nilai dalam peristiwa log untuk menambah nilai metrik

Anda dapat membuat filter metrik yang mempublikasikan nilai numerik yang ditemukan di peristiwa log Anda. Prosedur di bagian ini menggunakan contoh filter metrik berikut untuk menunjukkan bagaimana Anda dapat mempublikasikan nilai numerik dalam peristiwa log JSON ke metrik.

```
{ $.latency = * } metricValue: $.latency
```

Untuk membuat filter metrik yang menerbitkan nilai dalam peristiwa log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Log, lalu pilih Grup log.
3. Pilih atau buat grup log.

   Untuk informasi tentang cara membuat grup log, lihat Membuat grup log di CloudWatch Log di Panduan Pengguna CloudWatch Log Amazon.
4. Pilih Tindakan, lalu pilih Buat filter metrik.
5. Untuk Pola Filter**`{ $.latency = * }`**, masukkan, lalu pilih Berikutnya.
6. Untuk Nama Metrik, masukkan MyMetric.
7. Untuk Metric Value (Nilai Metrik), masukkan **`$.latency`**.

8. (Opsional) Untuk Nilai Default, masukkan 0, lalu pilih Berikutnya.

   Kami menyarankan Anda menentukan nilai default, bahkan jika nilainya 0. Menyetel nilai default membantu CloudWatch melaporkan data dengan lebih akurat dan CloudWatch mencegah agregasi metrik jerawatan. CloudWatch agregat dan melaporkan nilai metrik setiap menit.

9. Pilih Create metric filter (Buat filter metrik).

Filter metrik contoh cocok dengan istilah **`"latency"`** dalam contoh peristiwa log JSON dan menerbitkan nilai numerik 50 ke MyMetric metrik.

```
{
"latency": 50,
"requestType": "GET"
}
```

Membuat filter metrik

Prosedur dan contoh berikut menunjukkan cara membuat filter metrik.

Contoh

- Membuat filter metrik untuk grup log
- Contoh: Hitung peristiwa log
- Contoh: Hitung kemunculan suatu istilah
- Contoh: Hitung kode HTTP 404
- Contoh: Hitung kode HTTP 4xx
- Contoh: Mengekstraksi bidang dari log Apache dan menetapkan dimensi

Membuat filter metrik untuk grup log

Untuk membuat filter metrik untuk grup log, ikuti langkah-langkah ini. Metrik tidak akan terlihat sampai ada beberapa titik data untuk itu.

Untuk membuat filter metrik menggunakan CloudWatch konsol

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

2. Pada panel navigasi, pilih Log, lalu pilih Grup log.
3. Pilih nama grup log.
4. Pilih`Actions`, lalu pilih Buat filter metrik.
5. Untuk pola Filter, masukkan pola filter. Untuk informasi selengkapnya, lihat Filter sintaks pola untuk filter metrik, filter langganan, peristiwa log filter, dan Live Tail.
6. (Opsional) Untuk menguji pola filter Anda, di bawah Pola Uji, masukkan satu atau beberapa peristiwa log untuk menguji pola. Setiap peristiwa log harus diformat pada satu baris. Jeda baris digunakan untuk memisahkan peristiwa log di kotak pesan peristiwa Log.
7. Pilih Berikutnya, lalu masukkan nama untuk filter metrik Anda.
8. Di bawah Detail metrik, untuk namespace Metrik, masukkan nama untuk CloudWatch namespace tempat metrik akan dipublikasikan. Jika namespace belum ada, pastikan Create new dipilih.
9. Untuk Metric name (Nama metrik), masukkan nama untuk metrik baru.
10. Untuk Metric value (Nilai metrik), jika filter metrik Anda menghitung kemunculan kata kunci dalam filter, masukkan 1. Peningkatan akan menambahkan metrik dengan kelipatan sebesar 1 untuk setiap log acara yang mencakup salah satu kata kunci.

    Atau, masukkan token, seperti**`$size`**. Peningkatan ini akan menambahkan metrik sebesar nilai angka di bidang `size` untuk setiap log acara yang berisi bidang `size`.
11. (Opsional) Untuk Unit, pilih unit yang akan ditetapkan ke metrik. Jika Anda tidak menentukan unit, unit ditetapkan sebagai `None`.
12. (Opsional) Masukkan nama dan token untuk sebanyak tiga dimensi untuk metrik. Jika Anda menetapkan dimensi ke metrik yang dibuat oleh filter metrik, Anda tidak dapat menetapkan nilai default untuk metrik tersebut.

    > **Note**
    >
    > Dimensi hanya didukung di JSON atau filter metrik yang dibatasi ruang.
13. Pilih Create metric filter (Buat filter metrik). Anda dapat menemukan filter metrik yang Anda buat dari panel navigasi. Pilih Log, lalu pilih Grup log. Pilih nama grup log tempat Anda membuat filter metrik, lalu pilih tab Filter metrik.

Contoh: Hitung peristiwa log

Jenis paling sederhana dari pemantauan log acara adalah menghitung jumlah log acara yang terjadi. Anda mungkin ingin melakukan ini untuk menghitung jumlah semua kejadian, untuk membuat monitor gaya "detak jantung" atau hanya untuk berlatih membuat filter metrik.

Dalam contoh CLI berikut, filter metrik yang disebut MyAppAccessCount diterapkan ke grup log MyApp /access.log untuk membuat metrik EventCount di namespace. CloudWatch MyNamespace Filter dikonfigurasi untuk mencocokkan konten log acara dan menambahkan metrik dengan kelipatan sebesar "1".

Untuk membuat filter metrik menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Pilih nama grup log.
4. Pilih `Actions`, Create metric filter (Buat filter metrik).
5. Biarkan Filter Pattern (Pola Filter) dan Select Log Data to Test (Pilih Data Log untuk Pengujian) kosong.
6. Pilih Next (Selanjutnya), lalu untuk Filter Name (Nama Filter), ketik **`EventCount`**.
7. Di bawah Metric Details (Detail Metrik), untuk Metric Namespace, ketik **`MyNameSpace`**.
8. Untuk Metric Name (Nama Metrik), ketik **`MyAppEventCount`**.
9. Konfirmasi bahwa Metric Value (Nilai Metrik) adalah 1. Ini menentukan bahwa jumlah bertambah 1 untuk setiap log acara.
10. Masukkan 0 untuk Default Value (Nilai Default), lalu pilih Next (Selanjutnya). Menentukan nilai default memastikan bahwa data dilaporkan bahkan selama periode ketika tidak ada log acara terjadi sehingga mencegah metrik tidak teratur saat data terkadang tidak ada.
11. Pilih Create metric filter (Buat filter metrik).

Untuk membuat filter metrik menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
```

```
  --filter-pattern " " \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Anda dapat menguji kebijakan baru ini dengan memposting data kejadian apa pun. Anda akan melihat titik data yang dipublikasikan ke metrik MyAppAccessEventCount.

Untuk memposting data acara menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut:

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Contoh: Hitung kemunculan suatu istilah

Log acara sering mencakup pesan penting yang ingin Anda hitung, mungkin tentang keberhasilan atau kegagalan operasi. Sebagai contoh, kesalahan dapat terjadi dan dicatat ke berkas log jika operasi tertentu gagal. Anda mungkin ingin memantau entri ini untuk memahami tren kesalahan Anda.

Dalam contoh di bawah ini, filter metrik dibuat untuk memantau istilah Error. Kebijakan telah dibuat dan ditambahkan ke grup log MyApp/message.log. CloudWatch Log menerbitkan titik data ke metrik CloudWatch kustom ErrorCount di namespace MyApp/message.log dengan nilai “1" untuk setiap peristiwa yang berisi Kesalahan. Jika tidak ada kejadian berisi kata Error, nilai 0 akan dipublikasikan. Saat membuat grafik data ini di CloudWatch konsol, pastikan untuk menggunakan statistik penjumlahan.

Setelah membuat filter metrik, Anda dapat melihat metrik di CloudWatch konsol. Saat Anda memilih metrik yang akan ditampilkan, pilih namespace metrik yang cocok dengan nama grup log. Untuk informasi selengkapnya, lihat Melihat Metrik yang Tersedia.

Untuk membuat filter metrik menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.

3. Pilih nama grup log.
4. Pilih Actions (Tindakan), Create metric filter (Buat filter metrik).
5. Untuk Filter Pattern (Pola Filter), masukkan **`Error`**.

> **Note**
>
> Semua entri di Filter Pattern (Pola Filter) peka huruf besar-kecil.

6. (Opsional) Untuk menguji pola filter Anda, di Test Pattern (Pola Uji), masukkan satu atau beberapa log acara untuk digunakan menguji pola. Setiap log acara harus dalam satu baris, karena jeda baris yang digunakan untuk memisahkan log acara di kotak pesan log acara (Pesan log acara).
7. Pilih Next (Selanjutnya), lalu di halaman Assign metric (Tetapkan metrik), untuk Filter Name (Nama Filter), ketik **`MyAppErrorCount`**.
8. Di bawah Detail Metrik, untuk Ruang Nama Metrik, ketik. MyNameSpace
9. Untuk Metric Name (Nama Metrik), ketik ErrorCount.
10. Konfirmasi bahwa Metric Value (Nilai Metrik) adalah 1. Ini menentukan bahwa jumlah bertambah 1 untuk setiap log acara yang berisi "Error".
11. Untuk Default Value (Nilai Default) ketik 0, lalu pilih Next (Selanjutnya).
12. Pilih Create metric filter (Buat filter metrik).

Untuk membuat filter metrik menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut:

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Anda dapat menguji kebijakan baru ini dengan memposting kejadian yang berisi kata "Error" dalam pesannya.

Untuk memposting acara menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut. Perhatikan bahwa pola peka huruf besar dan kecil.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Contoh: Hitung kode HTTP 404

Menggunakan CloudWatch Log, Anda dapat memantau berapa kali server Apache Anda mengembalikan respons HTTP 404, yang merupakan kode respons untuk halaman yang tidak ditemukan. Anda mungkin ingin memantau ini untuk memahami seberapa sering pengunjung situs Anda tidak menemukan sumber daya yang mereka cari. Asumsikan bahwa struktur catatan log Anda menyertakan informasi berikut untuk setiap log acara (kunjungan situs):

- Alamat IP Peminta
- Identitas RFC 1413
- Nama pengguna
- Stempel waktu
- Metode permintaan dengan protokol dan sumber daya yang diminta
- Kode respons HTTP terhadap permintaan
- Byte yang ditransfer dalam permintaan

Contohnya dapat terlihat seperti berikut:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

Anda dapat menentukan aturan yang mencoba untuk mencocokkan kejadian dengan struktur seperti itu untuk kesalahan HTTP 404, seperti yang ditunjukkan dalam contoh berikut:

Untuk membuat filter metrik menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.

3. Pilih `Actions`, Create metric filter (Buat filter metrik).
4. Untuk Filter Pattern (Pola Filter), ketik **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**.
5. (Opsional) Untuk menguji pola filter Anda, di Test Pattern (Pola Uji), masukkan satu atau beberapa log acara untuk digunakan menguji pola. Setiap log acara harus dalam satu baris, karena jeda baris yang digunakan untuk memisahkan log acara di kotak pesan log acara (Pesan log acara).
6. Pilih Next (Selanjutnya), lalu untuk Filter Name (Nama Filter), ketik HTTP404Errors.
7. Di bawah Metric Details (Detail Metrik), untuk Metric Namespace (Namespace Metrik), masukkan **`MyNameSpace`**.
8. Untuk Metric Name (Nama Metrik), masukkan **`ApacheNotFoundErrorCount`**.
9. Konfirmasi bahwa Metric Value (Nilai Metrik) adalah 1. Ini menentukan bahwa jumlah bertambah 1 untuk setiap kejadian 404 Error.
10. Masukkan 0 untuk Default Value (Nilai Default), lalu pilih Next (Selanjutnya).
11. Pilih Create metric filter (Buat filter metrik).

Untuk membuat filter metrik menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
      metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

Dalam contoh ini, digunakan karakter literal, seperti tanda kurung siku kiri dan kanan, tanda kutip ganda, dan string karakter 404. Pola harus cocok dengan seluruh pesan log acara agar log acara dipertimbangkan untuk pemantauan.

Anda dapat memverifikasi pembuatan filter metrik dengan menggunakan perintah describe-metric-filters. Anda akan melihat output seperti ini:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

```
{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

Sekarang Anda dapat memposting beberapa kejadian secara manual:

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

Segera setelah meletakkan contoh peristiwa log ini, Anda dapat mengambil metrik yang dinamai di CloudWatch konsol sebagai ApacheNotFoundErrorCount.

Contoh: Hitung kode HTTP 4xx

Seperti dalam contoh sebelumnya, Anda mungkin ingin memantau log akses layanan web Anda dan memantau tingkat kode respons HTTP. Misalnya, Anda mungkin ingin memantau semua kesalahan HTTP di tingkat 400. Namun, Anda mungkin tidak ingin menentukan filter metrik baru untuk setiap kode yang dihasilkan.

Contoh berikut menunjukkan cara membuat metrik yang mencakup semua respons kode HTTP di tingkat 400 dari log akses menggunakan format log akses Apache dari contoh Contoh: Hitung kode HTTP 404.

Untuk membuat filter metrik menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Pilih nama grup log untuk server Apache.
4. Pilih `Actions`, Create metric filter (Buat filter metrik).
5. Untuk Filter pattern (Pola filter), masukkan **`[ip, id, user, timestamp, request, status_code=4*, size]`**.
6. (Opsional) Untuk menguji pola filter Anda, di Test Pattern (Pola Uji), masukkan satu atau beberapa log acara untuk digunakan menguji pola. Setiap log acara harus dalam satu baris, karena jeda baris yang digunakan untuk memisahkan log acara di kotak pesan log acara (Pesan log acara).
7. Pilih Next (Selanjutnya), lalu untuk Filter name (Nama filter), ketik **`HTTP4xxErrors`**.
8. Di bawah Metric details (Detail metrik), untuk Metric namespace (Namespace metrik), masukkan **`MyNameSpace`**.
9. Untuk Metric name (Nama metrik), masukkan HTTP4xxErrors.
10. Untuk Metric value (Nilai metrik), masukkan 1. Ini menentukan bahwa jumlah bertambah 1 untuk setiap log acara yang berisi kesalahan 4xx.
11. Masukkan 0 untuk Default value (Nilai default), lalu pilih Next (Selanjutnya).
12. Pilih Create metric filter (Buat filter metrik).

Untuk membuat filter metrik menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
  metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Anda dapat menggunakan data berikut dalam panggilan put-event untuk menguji aturan ini. Jika Anda tidak menghapus aturan pemantauan di contoh sebelumnya, Anda akan menghasilkan dua metrik yang berbeda.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Contoh: Mengekstraksi bidang dari log Apache dan menetapkan dimensi

Kadang-kadang, alih-alih menghitung, akan lebih berguna jika Anda menggunakan nilai dalam log acara individual untuk nilai metrik. Contoh ini menunjukkan cara Anda dapat membuat aturan ekstraksi untuk membuat metrik yang mengukur byte yang ditransfer oleh server web Apache.

Contoh ini juga menunjukkan cara menetapkan dimensi ke metrik yang Anda buat.

Untuk membuat filter metrik menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Pilih nama grup log untuk server Apache.
4. Pilih `Actions`, Create metric filter (Buat filter metrik).
5. Untuk Filter pattern (Pola filter), masukkan **`[ip, id, user, timestamp, request, status_code, size]`**.
6. (Opsional) Untuk menguji pola filter Anda, di Test Pattern (Pola Uji), masukkan satu atau beberapa log acara untuk digunakan menguji pola. Setiap log acara harus dalam satu baris, karena jeda baris yang digunakan untuk memisahkan log acara di kotak pesan log acara (Pesan log acara).
7. Pilih Next (Selanjutnya), lalu untuk Filter name (Nama filter), ketik **`size`**.
8. Di bawah Metric details (Detail metrik), untuk Metric namespace (Namespace metrik), masukkan **`MyNameSpace`**. Karena ini adalah namespace baru, pastikan bahwa Create new (Buat baru) dipilih.
9. Untuk Metric name (Nama metrik), masukkan **`BytesTransferred`**
10. Untuk Metric value (Nilai metrik), masukkan **`$size`**.
11. Untuk Unit, pilih Byte.
12. Untuk Dimension Name (Nama Dimensi), ketik **`IP`**.
13. Untuk Dimension Value (Nilai Dimensi), ketik **`$ip`**, lalu pilih Next (Selanjutnya).

14. Pilih Create metric filter (Buat filter metrik).

Untuk membuat filter metrik ini menggunakan AWS CLI

Di jendela perintah, jalankan perintah berikut

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
 --filter-name BytesTransferred \
 --filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
 --metric-transformations  \
 metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size'
```

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations  \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size',unit=Bytes,dimensio
$ip}}'
```

> **Note**
>
> Dalam perintah ini, gunakan format ini untuk menentukan beberapa dimensi.
>
> ```
> aws logs put-metric-filter \
> --log-group-name my-log-group-name \
> --filter-name my-filter-name \
> --filter-pattern 'my-filter-pattern' \
> --metric-transformations  \
> metricName=my-metric-name,metricNamespace=my-metric-namespace,metricValue=my-
> token,unit=unit,dimensions='{dimension1=$dim,dimension2=$dim2,dim3=$dim3}'
> ```

Anda dapat menggunakan data berikut dalam put-log-event panggilan untuk menguji aturan ini. Ini akan menghasilkan dua metrik yang berbeda jika Anda tidak menghapus aturan pemantauan dalam contoh sebelumnya.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
```

```
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Daftar filter metrik

Anda dapat membuat daftar semua filter metrik dalam grup log.

Untuk membuat daftar filter metrik menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Di panel konten, di daftar grup log, di kolom Metric Filters (Filter Metrik), pilih jumlah filter.

   Layar Log Groups > Filters for (Grup Log > Filter untuk) mencantumkan semua filter metrik yang terkait dengan grup log.

Untuk membuat daftar filter metrik menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

Berikut ini adalah output contoh:

```
{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
```

```
        }
    ]
}
```

Menghapus filter metrik

Kebijakan diidentifikasi berdasarkan nama dan grup lognya.

Untuk menghapus filter metrik menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Di panel konten, di kolom Metric Filter (Filter Metrik), pilih jumlah filter metrik untuk grup log.
4. Di layar Metric Filters (Filter Metrik), centang kotak di sebelah kanan nama filter yang ingin Anda hapus. Lalu pilih Hapus.
5. Saat diminta konfirmasi, pilih Hapus.

Untuk menghapus filter metrik menggunakan AWS CLI

Pada jendela perintah, jalankan perintah berikut:

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
 --filter-name MyFilterName
```

Pemrosesan data log secara real-time dengan langganan

Anda dapat menggunakan langganan untuk mendapatkan akses ke umpan real-time peristiwa CloudWatch log dari Log dan mengirimkannya ke layanan lain seperti aliran Amazon Kinesis, aliran Amazon Data Firehose, AWS Lambda atau untuk pemrosesan, analisis, atau pemuatan kustom ke sistem lain. Ketika peristiwa log dikirim ke layanan penerima, mereka dikodekan base64 dan dikompresi dengan format gzip.

Untuk mulai berlangganan peristiwa log, buat sumber daya penerima, seperti aliran Kinesis Data Streams, tempat acara akan dikirimkan. Filter langganan mendefinisikan pola filter yang akan digunakan untuk memfilter peristiwa log mana yang dikirim ke AWS sumber daya Anda, serta informasi tentang ke mana harus mengirim peristiwa log yang cocok.

Anda dapat membuat langganan di tingkat akun dan di tingkat grup log. Setiap akun dapat memiliki satu filter berlangganan tingkat akun. Setiap grup log dapat memiliki hingga dua filter langganan yang terkait dengan grup.

> **Note**
>
> Jika layanan tujuan mengembalikan kesalahan yang dapat dicoba ulang seperti pengecualian pembatasan atau pengecualian layanan yang dapat dicoba ulang (misalnya HTTP 5xx), CloudWatch Log terus mencoba lagi pengiriman hingga 24 jam. CloudWatch Log tidak mencoba mengirimkan kembali jika kesalahannya adalah kesalahan yang tidak dapat dicoba ulang, seperti atau. AccessDeniedException ResourceNotFoundException Dalam kasus ini, filter langganan dinonaktifkan hingga 10 menit, dan kemudian CloudWatch Log mencoba mengirim log ke tujuan. Selama periode dinonaktifkan ini, log dilewati.

CloudWatch Log juga menghasilkan CloudWatch metrik tentang penerusan peristiwa log ke langganan. Untuk informasi selengkapnya, lihat Pemantauan dengan CloudWatch metrik.

Anda juga dapat menggunakan langganan CloudWatch Log untuk mengalirkan data log dalam waktu dekat ke kluster OpenSearch Layanan Amazon. Untuk informasi selengkapnya, lihat data Streaming CloudWatch Log ke OpenSearch Layanan Amazon.

Langganan hanya didukung untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

> **Note**
>
> Filter langganan dapat mengumpulkan peristiwa log untuk mengoptimalkan transmisi dan mengurangi jumlah panggilan yang dilakukan ke tujuan. Batching tidak dijamin tetapi digunakan bila memungkinkan.

Daftar Isi

Konsep

Setiap filter langganan terdiri dari elemen kunci berikut:

pola filter

Deskripsi simbolis tentang bagaimana CloudWatch Log harus menafsirkan data di setiap peristiwa log, bersama dengan ekspresi pemfilteran yang membatasi apa yang dikirim ke sumber daya tujuan. AWS Untuk informasi selengkapnya tentang sintaks pola filter, lihat Filter sintaks pola untuk filter metrik, filter langganan, peristiwa log filter, dan Live Tail.

arn tujuan

Nama Sumber Daya Amazon (ARN) dari aliran Data Streams Kinesis, aliran Firehose, atau fungsi Lambda yang ingin Anda gunakan sebagai tujuan feed langganan.

arn peran

Peran IAM yang memberikan CloudWatch Log izin yang diperlukan untuk memasukkan data ke tujuan yang dipilih. Peran ini tidak diperlukan untuk tujuan Lambda karena CloudWatch Log bisa mendapatkan izin yang diperlukan dari pengaturan kontrol akses pada fungsi Lambda itu sendiri.

distribusi

Metode yang digunakan untuk mendistribusikan data log ke tujuan, ketika tujuan adalah aliran di Amazon Kinesis Data Streams. Secara default, data log dikelompokkan berdasarkan pengaliran log. Untuk distribusi yang lebih merata, Anda dapat mengelompokkan data log secara acak.

Untuk langganan tingkat grup log, elemen kunci berikut juga disertakan:

nama grup log

Grup log yang dikaitkan dengan filter langganan. Semua log acara yang diunggah ke grup log ini akan dikenakan filter langganan, dan log acara yang cocok dengan filter akan dikirim ke layanan tujuan yang menerima log acara yang cocok.

Untuk langganan tingkat akun, elemen kunci berikut juga disertakan:

kriteria seleksi

Kriteria yang digunakan untuk memilih grup log mana yang menerapkan filter langganan tingkat akun. Jika Anda tidak menentukan ini, filter langganan tingkat akun diterapkan ke semua grup log di akun. Bidang ini digunakan untuk mencegah loop log tak terbatas.. Untuk informasi selengkapnya tentang masalah loop log tak terbatas, lihatPencegahan rekursi log.

Kriteria seleksi memiliki batas ukuran 25 KB.

Filter langganan tingkat grup log

Anda dapat menggunakan filter langganan dengan Kinesis Data Streams, Lambda, atau Firehose. Log yang dikirim ke layanan penerima melalui filter langganan dikodekan base64 dan dikompresi dengan format gzip.

Anda dapat mencari data log Anda menggunakan sintaks Filter dan pola.

Contoh

- Contoh 1: Filter berlangganan dengan Kinesis Data Streams
- Contoh 2: Filter berlangganan dengan AWS Lambda
- Contoh 3: Filter berlangganan dengan Amazon Data Firehose

Contoh 1: Filter berlangganan dengan Kinesis Data Streams

Contoh berikut mengaitkan filter langganan dengan grup log yang berisi AWS CloudTrail peristiwa. Filter langganan mengirimkan setiap aktivitas yang dicatat yang dibuat oleh AWS kredenal “Root” ke aliran di Kinesis Data Streams yang disebut "." RootAccess Untuk informasi selengkapnya tentang cara mengirim AWS CloudTrail peristiwa ke CloudWatch Log, lihat Mengirim CloudTrail Acara ke CloudWatch Log di Panduan AWS CloudTrail Pengguna.

> **Note**
>
> Sebelum Anda membuat aliran, hitung volume data log yang akan dihasilkan. Pastikan untuk membuat aliran dengan pecahan yang cukup untuk menangani volume ini. Jika pengaliran tidak memiliki serpihan yang cukup, pengaliran log akan mengalami throttling. Untuk informasi selengkapnya tentang batas volume streaming, lihat Kuota dan Batas. Kiriman yang dibatasi dicoba ulang hingga 24 jam. Setelah 24 jam, kiriman yang gagal dijatuhkan.
> Untuk mengurangi risiko pelambatan, Anda dapat mengambil langkah-langkah berikut:
>
> - Tentukan `random distribution` kapan Anda membuat filter langganan dengan PutSubscriptionFilteratau put-subscription-filter. Secara default, distribusi filter aliran adalah dengan aliran log dan ini dapat menyebabkan pelambatan.
> - Pantau streaming Anda menggunakan CloudWatch metrik. Ini membantu Anda mengidentifikasi pelambatan apa pun dan menyesuaikan konfigurasi Anda sesuai dengan itu. Misalnya, `DeliveryThrottling` metrik dapat digunakan untuk melacak jumlah peristiwa CloudWatch log yang Log dibatasi saat meneruskan data ke tujuan langganan. Untuk informasi selengkapnya tentang pemantauan, lihat Pemantauan dengan CloudWatch metrik.
> - Gunakan mode kapasitas sesuai permintaan untuk streaming Anda di Kinesis Data Streams. Mode on-demand langsung mengakomodasi beban kerja Anda saat mereka naik atau turun. Informasi selengkapnya tentang mode kapasitas sesuai permintaan, lihat Mode sesuai permintaan.
> - Batasi pola filter CloudWatch langganan Anda agar sesuai dengan kapasitas streaming Anda di Kinesis Data Streams. Jika Anda mengirim terlalu banyak data ke aliran, Anda mungkin perlu mengurangi ukuran filter atau menyesuaikan kriteria filter.

Untuk membuat filter langganan untuk Kinesis Data Streams

1. Buat aliran tujuan menggunakan perintah berikut:

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. Tunggu hingga aliran menjadi Aktif (ini mungkin memakan waktu satu atau dua menit). Anda dapat menggunakan perintah Kinesis Data Streams describe-stream berikut untuk memeriksa. StreamDescription StreamStatusproperti. Selain itu, perhatikan StreamDescriptionnilai.streaMarn, karena Anda akan membutuhkannya di langkah selanjutnya:

```
aws kinesis describe-stream --stream-name "RootAccess"
```

   Berikut ini adalah output contoh:

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "49551135218688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Buat peran IAM yang akan memberikan izin CloudWatch Log untuk memasukkan data ke aliran Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file (misalnya, `~/TrustPolicyForCWL-Kinesis.json`). Gunakan editor teks untuk membuat kebijakan ini. Jangan gunakan konsol IAM untuk membuatnya.

Kebijakan ini mencakup kunci konteks kondisi `aws:SourceArn` global untuk membantu mencegah masalah keamanan wakil yang membingungkan. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. Gunakan perintah create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Perhatikan nilai Role.Arn yang dihasilkan, karena Anda juga akan membutuhkannya untuk langkah selanjutnya:

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

Berikut adalah contoh output.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
```

```
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Buat kebijakan izin untuk menentukan tindakan apa yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, Anda akan membuat kebijakan izin dalam file (misalnya, `~/PermissionsForCWL-Kinesis.json`). Gunakan editor teks untuk membuat kebijakan ini. Jangan gunakan konsol IAM untuk membuatnya.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
  ]
}
```

6. Kaitkan kebijakan izin dengan peran menggunakan put-role-policyperintah berikut:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Setelah streaming dalam status Aktif dan Anda telah membuat peran IAM, Anda dapat membuat filter langganan CloudWatch Log. Filter langganan segera memulai aliran data log waktu nyata dari grup log yang dipilih ke aliran Anda:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. Setelah Anda mengatur filter langganan, CloudWatch Log meneruskan semua peristiwa log masuk yang cocok dengan pola filter ke aliran Anda. Anda dapat memverifikasi bahwa ini terjadi dengan mengambil iterator pecahan Kinesis Data Streams dan menggunakan perintah Kinesis Data Streams get-records untuk mengambil beberapa catatan Kinesis Data Streams:

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

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Perhatikan bahwa Anda mungkin perlu melakukan panggilan ini beberapa kali sebelum Kinesis Data Streams mulai mengembalikan data.

Anda akan melihat respons dengan array catatan. Atribut Data dalam catatan Kinesis Data Streams adalah base64 dikodekan dan dikompresi dengan format gzip. Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Data yang didekode dan didekompresi base64 diformat sebagai JSON dengan struktur berikut:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
```

```
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```

Elemen kunci dalam struktur data di atas adalah sebagai berikut:

owner

ID AWS Akun dari data log asal.

logGroup

Nama grup log dari data log asal.

logStream

Nama pengaliran log dari data log asal.

subscriptionFilters

Daftar nama filter langganan yang cocok dengan data log asal.

messageType

Pesan data akan menggunakan tipe "DATA_MESSAGE". Terkadang CloudWatch Log dapat memancarkan catatan Kinesis Data Streams dengan tipe “CONTROL_MESSAGE”, terutama untuk memeriksa apakah tujuan dapat dijangkau.

logEvents

Data log yang sebenarnya, direpresentasikan sebagai array catatan log acara. Properti "id" adalah pengenal unik untuk setiap log acara.

Contoh 2: Filter berlangganan dengan AWS Lambda

Dalam contoh ini, Anda akan membuat filter langganan CloudWatch Log yang mengirimkan data log ke AWS Lambda fungsi Anda.

> **Note**
>
> Sebelum membuat fungsi Lambda, hitung volume data log yang akan dihasilkan. Pastikan untuk membuat fungsi yang dapat menangani volume ini. Jika fungsi tidak memiliki volume yang cukup, pengaliran log akan mengalami throttling. Untuk informasi selengkapnya tentang batas Lambda, lihat Batas AWS Lambda.

Untuk membuat filter langganan untuk Lambda

1. Buat AWS Lambda fungsinya.

   Pastikan bahwa Anda telah mengatur peran eksekusi Lambda. Untuk informasi selengkapnya, lihat: Langkah 2.2: Buat IAM role (peran eksekusi) dalam Panduan Developer AWS Lambda .

2. Buka editor teks dan buat file bernama `helloWorld.js`dengan isi sebagai berikut:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Buat zip file helloWorld.js dan simpan dengan nama `helloWorld.zip`.
4. Gunakan perintah berikut, di mana perannya adalah peran eksekusi Lambda yang Anda atur di langkah pertama:

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. Berikan CloudWatch Log izin untuk menjalankan fungsi Anda. Gunakan perintah berikut, dengan mengganti akun placeholder dengan akun Anda sendiri dan grup log placeholder dengan grup log yang akan diproses:

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. Buat filter langganan menggunakan perintah berikut, dengan mengganti akun placeholder dengan akun Anda sendiri dan grup log placeholder dengan grup log yang akan diproses:

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. (Opsional) Uji menggunakan contoh log acara. Di jendela perintah, jalankan perintah berikut, yang akan menempatkan pesan log sederhana ke dalam pengaliran langganan.

   Untuk melihat output dari fungsi Lambda Anda, buka fungsi Lambda dan Anda akan melihat output di /aws/lambda/helloworld:

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --
log-events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple
 Lambda Test\"}]"
```

Anda akan melihat respons dengan array Lambda. Atribut Data dalam catatan Lambda adalah base64 dikodekan dan dikompresi dengan format gzip. Muatan sebenarnya yang diterima oleh Lambda memiliki format berikut `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }` Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

Data yang didekode dan didekompresi base64 diformat sebagai JSON dengan struktur berikut:

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```

Elemen kunci dalam struktur data di atas adalah sebagai berikut:

owner

ID AWS Akun dari data log asal.

logGroup

Nama grup log dari data log asal.

logStream

Nama pengaliran log dari data log asal.

subscriptionFilters

Daftar nama filter langganan yang cocok dengan data log asal.

messageType

Pesan data akan menggunakan tipe "DATA_MESSAGE". Terkadang CloudWatch Log dapat memancarkan catatan Lambda dengan tipe “CONTROL_MESSAGE”, terutama untuk memeriksa apakah tujuan dapat dijangkau.

logEvents

Data log yang sebenarnya, direpresentasikan sebagai array catatan log acara. Properti "id" adalah pengenal unik untuk setiap log acara.

Contoh 3: Filter berlangganan dengan Amazon Data Firehose

Dalam contoh ini, Anda akan membuat langganan CloudWatch Log yang mengirimkan peristiwa log masuk yang cocok dengan filter yang ditentukan ke aliran pengiriman Amazon Data Firehose. Data yang dikirim dari CloudWatch Log ke Amazon Data Firehose sudah dikompresi dengan kompresi gzip level 6, jadi Anda tidak perlu menggunakan kompresi dalam aliran pengiriman Firehose Anda. Anda kemudian dapat menggunakan fitur dekompresi di Firehose untuk mendekompresi log secara otomatis. Untuk informasi selengkapnya, lihat Menulis ke Kinesis Data CloudWatch Firehose Menggunakan Log.

> **Note**
>
> Sebelum Anda membuat aliran Firehose, hitung volume data log yang akan dihasilkan. Pastikan untuk membuat aliran Firehose yang dapat menangani volume ini. Jika pengaliran

tidak dapat menangani volume, pengaliran log akan mengalami throttling. Untuk informasi selengkapnya tentang batas volume aliran Firehose, lihat Batas Data Firehose Amazon Data.

Untuk membuat filter langganan untuk Firehose

1. Buat bucket Amazon Simple Storage Service (Amazon S3). Kami menyarankan Anda menggunakan bucket yang dibuat khusus untuk CloudWatch Log. Namun, jika Anda ingin menggunakan bucket yang sudah ada, lewati ke langkah 2.

   Jalankan perintah berikut, dengan mengganti Wilayah placeholder dengan Wilayah yang ingin Anda gunakan:

   ```
   aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
    LocationConstraint=region
   ```

   Berikut ini adalah output contoh:

   ```
   {
       "Location": "/my-bucket"
   }
   ```

2. Buat peran IAM yang memberikan izin Amazon Data Firehose untuk memasukkan data ke dalam bucket Amazon S3 Anda.

   Untuk informasi selengkapnya, lihat Mengontrol Akses dengan Amazon Data Firehose di Panduan Pengembang Amazon Data Firehose.

   Pertama, gunakan editor teks untuk membuat kebijakan kepercayaan dalam file `~/TrustPolicyForFirehose.json` sebagai berikut:

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "firehose.amazonaws.com" },
       "Action": "sts:AssumeRole"
       }
   }
   ```

3. Gunakan perintah create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Perhatikan nilai Role.Arn yang dihasilkan, karena Anda akan membutuhkannya dalam langkah selanjutnya:

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Buat kebijakan izin untuk menentukan tindakan apa yang dapat dilakukan Firehose di akun Anda. Pertama, gunakan editor teks untuk membuat kebijakan izin dalam file *~/*`PermissionsForFirehose.json`:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
```

```
      "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. Kaitkan kebijakan izin dengan peran menggunakan put-role-policy perintah berikut:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Buat aliran pengiriman Firehose tujuan sebagai berikut, ganti nilai placeholder untuk RoLearn dan BuckEtarn dengan ARN peran dan bucket yang Anda buat:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

   Perhatikan bahwa Firehose secara otomatis menggunakan awalan dalam format waktu UTC YYYYY/MM/DD/HH untuk objek Amazon S3 yang dikirimkan. Anda dapat menentukan prefiks tambahan untuk ditambahkan di depan prefiks format waktu. Jika prefiks berakhir dengan garis miring (/), itu akan muncul sebagai folder dalam bucket Amazon S3.

7. Tunggu sampai pengaliran menjadi aktif (ini mungkin memakan waktu beberapa menit). Anda dapat menggunakan describe-delivery-streamperintah Firehose untuk memeriksa. DeliveryStreamDescription DeliveryStreamStatusproperti. Selain itu, perhatikan DeliveryStreamDescription. DeliveryStreamNilai ARN, karena Anda akan membutuhkannya di langkah selanjutnya:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
```

```
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Buat peran IAM yang memberikan izin CloudWatch Log untuk memasukkan data ke aliran pengiriman Firehose Anda. Pertama, gunakan editor teks untuk membuat kebijakan kepercayaan dalam file `~/TrustPolicyForCWL.json`:

   Kebijakan ini mencakup kunci konteks kondisi `aws:SourceArn` global untuk membantu mencegah masalah keamanan wakil yang membingungkan. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
  }
}
```

9. Gunakan perintah create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Perhatikan nilai Role.Arn yang dihasilkan, karena Anda akan membutuhkannya dalam langkah selanjutnya:

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Buat kebijakan izin untuk menentukan tindakan apa yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, gunakan editor teks untuk membuat file kebijakan izin (misalnya, `~/PermissionsForCWL.json`):

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
```

```
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Kaitkan kebijakan izin dengan peran menggunakan put-role-policy perintah:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Setelah aliran pengiriman Amazon Data Firehose dalam keadaan aktif dan Anda telah membuat peran IAM, Anda dapat membuat filter langganan CloudWatch Log. Filter langganan segera memulai aliran data log waktu nyata dari grup log yang dipilih ke aliran pengiriman Amazon Data Firehose Anda:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. Setelah Anda mengatur filter langganan, CloudWatch Log akan meneruskan semua peristiwa log masuk yang cocok dengan pola filter ke aliran pengiriman Amazon Data Firehose Anda. Data Anda akan mulai muncul di Amazon S3 berdasarkan interval buffer waktu yang ditetapkan pada aliran pengiriman Amazon Data Firehose Anda. Setelah waktu tertentu berlalu, Anda dapat memverifikasi data dengan memeriksa Bucket Amazon S3 Anda.

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
```

```
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2015/10/29/00/my-
delivery-stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Data dalam objek Amazon S3 dikompresi dengan format gzip. Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

```
zcat testfile.gz
```

Filter berlangganan tingkat akun

> **Important**
>
> Ada risiko menyebabkan loop rekursif tak terbatas dengan filter berlangganan yang dapat menyebabkan peningkatan besar dalam penagihan konsumsi jika tidak ditangani. Untuk mengurangi risiko ini, sebaiknya gunakan kriteria pemilihan di filter langganan tingkat akun untuk mengecualikan grup log yang menyerap data log dari sumber daya yang merupakan bagian dari alur kerja pengiriman langganan. Untuk informasi selengkapnya tentang masalah ini dan menentukan grup log mana yang akan dikecualikan, lihatPencegahan rekursi log.

Anda dapat menetapkan kebijakan berlangganan tingkat akun yang mencakup subset grup log di akun. Kebijakan berlangganan akun dapat bekerja dengan Kinesis Data Streams, Lambda, atau Firehose. Log yang dikirim ke layanan penerima melalui kebijakan berlangganan tingkat akun dikodekan base64 dan dikompresi dengan format gzip.

> **Note**
>
> Untuk melihat daftar semua kebijakan filter langganan di akun Anda, gunakan `describe-account-policies` perintah dengan nilai `SUBSCRIPTION_FILTER_POLICY` `--policy-type` parameter. Untuk informasi lebih lanjut, lihat describe-account-policies¶.

Contoh

- Contoh 1: Filter berlangganan dengan Kinesis Data Streams
- Contoh 2: Filter berlangganan dengan AWS Lambda
- Contoh 3: Filter berlangganan dengan Amazon Data Firehose

Contoh 1: Filter berlangganan dengan Kinesis Data Streams

Sebelum Anda membuat aliran data Kinesis Data Streams untuk digunakan dengan kebijakan langganan tingkat akun, hitung volume data log yang akan dihasilkan. Pastikan untuk membuat aliran dengan pecahan yang cukup untuk menangani volume ini. Jika aliran tidak memiliki cukup pecahan, itu dibatasi. Untuk informasi selengkapnya tentang batas volume stream, lihat Kuota dan Batas dalam dokumentasi Kinesis Data Streams.

> **Warning**
>
> Karena peristiwa log dari beberapa grup log diteruskan ke tujuan, ada risiko pembatasan. Kiriman yang dibatasi dicoba ulang hingga 24 jam. Setelah 24 jam, kiriman yang gagal dijatuhkan.
> Untuk mengurangi risiko pelambatan, Anda dapat mengambil langkah-langkah berikut:
>
> - Pantau aliran Kinesis Data Streams CloudWatch Anda dengan metrik. Ini membantu Anda mengidentifikasi pelambatan dan menyesuaikan konfigurasi Anda sesuai dengan itu. Misalnya, `DeliveryThrottling` metrik melacak jumlah peristiwa log yang CloudWatch Log dibatasi saat meneruskan data ke tujuan langganan. Untuk informasi selengkapnya, lihat Pemantauan dengan CloudWatch metrik.
> - Gunakan mode kapasitas sesuai permintaan untuk streaming Anda di Kinesis Data Streams. Mode on-demand langsung mengakomodasi beban kerja Anda saat mereka naik atau turun. Untuk informasi selengkapnya, lihat Mode sesuai permintaan.
> - Batasi pola filter langganan CloudWatch Log agar sesuai dengan kapasitas streaming Anda di Kinesis Data Streams. Jika Anda mengirim terlalu banyak data ke aliran, Anda mungkin perlu mengurangi ukuran filter atau menyesuaikan kriteria filter.

Contoh berikut menggunakan kebijakan berlangganan tingkat akun untuk meneruskan semua peristiwa log ke aliran di Kinesis Data Streams. Pola filter cocok dengan setiap peristiwa log dengan teks `Test` dan meneruskannya ke aliran di Kinesis Data Streams.

Membuat kebijakan berlangganan tingkat akun untuk Kinesis Data Streams

1. Buat aliran tujuan menggunakan perintah berikut:

```
$ C:\>  aws kinesis create-stream –stream-name "TestStream" –shard-count 1
```

2. Tunggu beberapa menit hingga streaming menjadi aktif. Anda dapat memverifikasi apakah aliran aktif dengan menggunakan perintah describe-stream untuk memeriksa. StreamDescription StreamStatusproperti.

```
aws kinesis describe-stream --stream-name "TestStream"
```

   Berikut ini adalah output contoh:

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "TestStream",
        "StreamARN": "arn:aws:kinesis:region:123456789012:stream/TestStream",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "EXAMPLE84634633746074317682114555",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "EXAMPLE6888184566795038319814587845913527021815726l0"
                }
            }
        ]
    }
}
```

3. Buat peran IAM yang akan memberikan izin CloudWatch Log untuk memasukkan data ke aliran Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file (misalnya, `~/TrustPolicyForCWL-Kinesis.json`). Gunakan editor teks untuk membuat kebijakan ini.

   Kebijakan ini mencakup kunci konteks kondisi `aws:SourceArn` global untuk membantu mencegah masalah keamanan wakil yang membingungkan. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. Gunakan perintah create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Perhatikan nilai Role.Arn yang dihasilkan, karena Anda juga akan membutuhkannya untuk langkah selanjutnya:

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

Berikut adalah contoh output.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "EXAMPLE450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Buat kebijakan izin untuk menentukan tindakan apa yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, Anda akan membuat kebijakan izin dalam file (misalnya, `~/PermissionsForCWL-Kinesis.json`). Gunakan editor teks untuk membuat kebijakan ini. Jangan gunakan konsol IAM untuk membuatnya.

```
{
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/TestStream"
    }
  ]
}
```

6. Kaitkan kebijakan izin dengan peran menggunakan put-role-policyperintah berikut:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Setelah aliran dalam status Aktif dan Anda telah membuat peran IAM, Anda dapat membuat kebijakan filter langganan CloudWatch Log. Kebijakan segera memulai aliran data log waktu nyata ke aliran Anda. Dalam contoh ini, semua peristiwa log yang berisi string dialirkan, kecuali yang ERROR ada di grup log bernama LogGroupToExclude1 danLogGroupToExclude2.

```
aws logs put-account-policy \
    --policy-name "ExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisRole", "DestinationArn":"arn:aws:kinesis:region:123456789012:stream/
TestStream", "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

8. Setelah menyiapkan filter langganan, CloudWatch Log meneruskan semua peristiwa log masuk yang cocok dengan pola filter dan kriteria pemilihan ke aliran Anda.

   `selection-criteria`Bidang ini opsional, tetapi penting untuk mengecualikan grup log yang dapat menyebabkan rekursi log tak terbatas dari filter langganan. Untuk informasi selengkapnya tentang masalah ini dan menentukan grup log mana yang akan dikecualikan, lihatPencegahan rekursi log. Saat ini, NOT IN adalah satu-satunya operator yang didukung untuk`selection-criteria`.

   Anda dapat memverifikasi bahwa alur peristiwa log dengan menggunakan iterator pecahan Kinesis Data Streams dan menggunakan `get-records` perintah Kinesis Data Streams untuk mengambil beberapa catatan Kinesis Data Streams:

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

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Anda mungkin perlu menggunakan perintah ini beberapa kali sebelum Kinesis Data Streams mulai mengembalikan data.

Anda akan melihat respons dengan array catatan. Atribut Data dalam catatan Kinesis Data Streams adalah base64 dikodekan dan dikompresi dengan format gzip. Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Data yang didekode dan didekompresi base64 diformat sebagai JSON dengan struktur berikut:

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicy"
    ],
    "logEvents": [
        {
```

```
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```


Elemen kunci dalam struktur data adalah sebagai berikut:

messageType

Pesan data akan menggunakan tipe "DATA_MESSAGE". Terkadang CloudWatch Log mungkin memancarkan catatan Kinesis Data Streams dengan tipe “CONTROL_MESSAGE”, terutama untuk memeriksa apakah tujuan dapat dijangkau.

owner

ID AWS Akun dari data log asal.

logGroup

Nama grup log dari data log asal.

logStream

Nama pengaliran log dari data log asal.

subscriptionFilters

Daftar nama filter langganan yang cocok dengan data log asal.

logEvents

Data log yang sebenarnya, direpresentasikan sebagai array catatan log acara. Properti "id" adalah pengenal unik untuk setiap log acara.

PolicyLevel

Tingkat di mana kebijakan itu ditegakkan. “ACCOUNT_LEVEL_POLICY” adalah `policyLevel` untuk kebijakan filter langganan tingkat akun.

Contoh 2: Filter berlangganan dengan AWS Lambda

Dalam contoh ini, Anda akan membuat kebijakan filter langganan tingkat akun CloudWatch Log yang mengirimkan data log ke fungsi Anda AWS Lambda .

> **Warning**
>
> Sebelum membuat fungsi Lambda, hitung volume data log yang akan dihasilkan. Pastikan untuk membuat fungsi yang dapat menangani volume ini. Jika fungsi tidak dapat menangani volume, aliran log akan dibatasi. Karena peristiwa log dari semua grup log atau subset grup log akun diteruskan ke tujuan, ada risiko pembatasan. Untuk informasi selengkapnya tentang batas Lambda, lihat Batas AWS Lambda.

Untuk membuat kebijakan filter langganan tingkat akun untuk Lambda

1. Buat AWS Lambda fungsinya.

   Pastikan bahwa Anda telah mengatur peran eksekusi Lambda. Untuk informasi selengkapnya, lihat: Langkah 2.2: Buat IAM role (peran eksekusi) dalam Panduan Developer AWS Lambda .
2. Buka editor teks dan buat file bernama `helloWorld.js`dengan isi sebagai berikut:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
```

```
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Buat zip file helloWorld.js dan simpan dengan nama `helloWorld.zip`.
4. Gunakan perintah berikut, di mana perannya adalah peran eksekusi Lambda yang Anda atur di langkah pertama:

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs18.x
```

5. Berikan CloudWatch Log izin untuk menjalankan fungsi Anda. Gunakan perintah berikut, ganti akun placeholder dengan akun Anda sendiri.

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789012:log-group:*" \
    --source-account "123456789012"
```

6. Buat kebijakan filter langganan tingkat akun menggunakan perintah berikut, ganti akun placeholder dengan akun Anda sendiri. Dalam contoh ini, semua peristiwa log yang berisi string dialirkan, kecuali yang ERROR ada di grup log bernama `LogGroupToExclude1` dan`LogGroupToExclude2`.

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyLambda" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:lambda:region:123456789012:function:helloWorld",
 "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
```

```
    --scope "ALL"
```

Setelah menyiapkan filter langganan, CloudWatch Log meneruskan semua peristiwa log masuk yang cocok dengan pola filter dan kriteria pemilihan ke aliran Anda.

`selection-criteria`Bidang ini opsional, tetapi penting untuk mengecualikan grup log yang dapat menyebabkan rekursi log tak terbatas dari filter langganan. Untuk informasi selengkapnya tentang masalah ini dan menentukan grup log mana yang akan dikecualikan, lihatPencegahan rekursi log. Saat ini, NOT IN adalah satu-satunya operator yang didukung untuk`selection-criteria`.

7. (Opsional) Uji menggunakan contoh log acara. Di jendela perintah, jalankan perintah berikut, yang akan menempatkan pesan log sederhana ke dalam pengaliran langganan.

   Untuk melihat output dari fungsi Lambda Anda, buka fungsi Lambda dan Anda akan melihat output di /aws/lambda/helloworld:

   ```
   aws logs put-log-events --log-group-name Example1 --log-stream-name logStream1 --
   log-events "[{\"timestamp\":CURRENT TIMESTAMP MILLIS , \"message\": \"Simple Lambda
    Test\"}]"
   ```

   Anda akan melihat respons dengan array Lambda. Atribut Data dalam catatan Lambda adalah base64 dikodekan dan dikompresi dengan format gzip. Muatan sebenarnya yang diterima oleh Lambda memiliki format berikut `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }` Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

   ```
   echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
   ```

   Data yang didekode dan didekompresi base64 diformat sebagai JSON dengan struktur berikut:

   ```
   {
       "messageType": "DATA_MESSAGE",
       "owner": "123456789012",
       "logGroup": "Example1",
       "logStream": "logStream1",
       "subscriptionFilters": [
           "ExamplePolicyLambda"
       ],
       "logEvents": [
   ```

```
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

> **Note**
>
> Filter langganan tingkat akun tidak akan diterapkan ke grup log fungsi Lambda tujuan. Ini untuk mencegah rekursi log tak terbatas yang dapat menyebabkan peningkatan penagihan konsumsi. Untuk informasi lebih lanjut tentang masalah ini, lihatPencegahan rekursi log.

Elemen kunci dalam struktur data adalah sebagai berikut:

messageType

Pesan data akan menggunakan tipe "DATA_MESSAGE". Terkadang CloudWatch Log mungkin memancarkan catatan Kinesis Data Streams dengan tipe “CONTROL_MESSAGE”, terutama untuk memeriksa apakah tujuan dapat dijangkau.

owner

ID AWS Akun dari data log asal.

logGroup

Nama grup log dari data log asal.

logStream

Nama pengaliran log dari data log asal.

subscriptionFilters

Daftar nama filter langganan yang cocok dengan data log asal.

logEvents

Data log yang sebenarnya, direpresentasikan sebagai array catatan log acara. Properti "id" adalah pengenal unik untuk setiap log acara.

PolicyLevel

Tingkat di mana kebijakan itu ditegakkan. “ACCOUNT_LEVEL_POLICY” adalah `policyLevel` untuk kebijakan filter langganan tingkat akun.

Contoh 3: Filter berlangganan dengan Amazon Data Firehose

Dalam contoh ini, Anda akan membuat kebijakan filter langganan tingkat akun CloudWatch Log yang mengirimkan peristiwa log masuk yang cocok dengan filter yang ditentukan ke aliran pengiriman Amazon Data Firehose. Data yang dikirim dari CloudWatch Log ke Amazon Data Firehose sudah dikompresi dengan kompresi gzip level 6, jadi Anda tidak perlu menggunakan kompresi dalam aliran pengiriman Firehose Anda. Anda kemudian dapat menggunakan fitur dekompresi di Firehose untuk mendekompresi log secara otomatis. Untuk informasi selengkapnya, lihat Menulis ke Kinesis Data CloudWatch Firehose Menggunakan Log.

> **Warning**
>
> Sebelum Anda membuat aliran Firehose, hitung volume data log yang akan dihasilkan. Pastikan untuk membuat aliran Firehose yang dapat menangani volume ini. Jika pengaliran tidak dapat menangani volume, pengaliran log akan mengalami throttling. Untuk informasi selengkapnya tentang batas volume aliran Firehose, lihat Batas Data Firehose Amazon Data.

Untuk membuat filter langganan untuk Firehose

1. Buat bucket Amazon Simple Storage Service (Amazon S3). Kami menyarankan Anda menggunakan bucket yang dibuat khusus untuk CloudWatch Log. Namun, jika Anda ingin menggunakan bucket yang sudah ada, lewati ke langkah 2.

   Jalankan perintah berikut, dengan mengganti Wilayah placeholder dengan Wilayah yang ingin Anda gunakan:

   ```
   aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
    LocationConstraint=region
   ```

   Berikut ini adalah output contoh:

   ```
   {
       "Location": "/my-bucket"
   }
   ```

2. Buat peran IAM yang memberikan izin Amazon Data Firehose untuk memasukkan data ke dalam bucket Amazon S3 Anda.

   Untuk informasi selengkapnya, lihat Mengontrol Akses dengan Amazon Data Firehose di Panduan Pengembang Amazon Data Firehose.

   Pertama, gunakan editor teks untuk membuat kebijakan kepercayaan dalam file `~/TrustPolicyForFirehose.json` sebagai berikut:

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "firehose.amazonaws.com" },
       "Action": "sts:AssumeRole"
       }
   }
   ```

3. Gunakan perintah create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Catat nilai Role.Arn yang dikembalikan, karena Anda akan membutuhkannya di langkah selanjutnya:

   ```
   aws iam create-role \
    --role-name FirehosetoS3Role \
   ```

```
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "EXAMPLE50GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Buat kebijakan izin untuk menentukan tindakan apa yang dapat dilakukan Firehose di akun Anda. Pertama, gunakan editor teks untuk membuat kebijakan izin dalam file *~/*`PermissionsForFirehose.json`:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. Kaitkan kebijakan izin dengan peran menggunakan put-role-policy perintah berikut:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Buat aliran pengiriman Firehose tujuan sebagai berikut, ganti nilai placeholder untuk RoLearn dan BuckEtarn dengan ARN peran dan bucket yang Anda buat:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

   NFirehose secara otomatis menggunakan awalan dalam format waktu YYYY/MM/DD/HH UTC untuk objek Amazon S3 yang dikirimkan. Anda dapat menentukan prefiks tambahan untuk ditambahkan di depan prefiks format waktu. Jika prefiks berakhir dengan garis miring (/), itu akan muncul sebagai folder dalam bucket Amazon S3.

7. Tunggu beberapa menit hingga streaming menjadi aktif. Anda dapat menggunakan describe-delivery-streamperintah Firehose untuk memeriksa. DeliveryStreamDescription DeliveryStreamStatusproperti. Selain itu, perhatikan DeliveryStreamDescription. DeliveryStreamNilai ARN, karena Anda akan membutuhkannya di langkah selanjutnya:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
```

```
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Buat peran IAM yang memberikan izin CloudWatch Log untuk memasukkan data ke aliran pengiriman Firehose Anda. Pertama, gunakan editor teks untuk membuat kebijakan kepercayaan dalam file ~/TrustPolicyForCWL.json:

   Kebijakan ini mencakup kunci konteks kondisi aws:SourceArn global untuk membantu mencegah masalah keamanan wakil yang membingungkan. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
  }
}
```

9. Gunakan perintah create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Catat nilai Role.Arn yang dikembalikan, karena Anda akan membutuhkannya di langkah selanjutnya:

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json
```

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Buat kebijakan izin untuk menentukan tindakan apa yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, gunakan editor teks untuk membuat file kebijakan izin (misalnya, `~/PermissionsForCWL.json`):

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-name"]
      }
    ]
}
```

11. Kaitkan kebijakan izin dengan peran menggunakan put-role-policy perintah:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Setelah aliran pengiriman Amazon Data Firehose dalam status aktif dan Anda telah membuat peran IAM, Anda dapat membuat kebijakan filter langganan tingkat akun CloudWatch Log. Kebijakan segera memulai aliran data log real-time dari grup log yang dipilih ke aliran pengiriman Amazon Data Firehose Anda:

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyFirehose" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisFirehoseRole", "DestinationArn":"arn:aws:firehose:us-
east-1:123456789012:deliverystream/delivery-stream-name", "FilterPattern": "Test",
 "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

13. Setelah menyiapkan filter langganan, CloudWatch Log meneruskan peristiwa log masuk yang cocok dengan pola filter ke aliran pengiriman Amazon Data Firehose Anda.

    `selection-criteria`Bidang ini opsional, tetapi penting untuk mengecualikan grup log yang dapat menyebabkan rekursi log tak terbatas dari filter langganan. Untuk informasi selengkapnya tentang masalah ini dan menentukan grup log mana yang akan dikecualikan, lihatPencegahan rekursi log. Saat ini, NOT IN adalah satu-satunya operator yang didukung untuk`selection-criteria`.

    Data Anda akan mulai muncul di Amazon S3 berdasarkan interval buffer waktu yang ditetapkan pada aliran pengiriman Amazon Data Firehose Anda. Setelah waktu tertentu berlalu, Anda dapat memverifikasi data dengan memeriksa Bucket Amazon S3 Anda.

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2023-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
```

```
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2023/10/29/00/my-delivery-stream-2023-10-29-00-35-40-
EXAMPLE-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "EXAMPLE6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2023/10/29/00/my-
delivery-stream-2023-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2023 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Data dalam objek Amazon S3 dikompresi dengan format gzip. Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

```
zcat testfile.gz
```

Langganan lintas akun Lintas wilayah

Anda dapat berkolaborasi dengan pemilik AWS akun lain dan menerima peristiwa log mereka di AWS sumber daya Anda, seperti Amazon Kinesis atau Amazon Data Firehose stream (ini dikenal sebagai berbagi data lintas akun). Misalnya, data peristiwa log ini dapat dibaca dari Aliran Data Kinesis terpusat atau aliran Firehose untuk melakukan pemrosesan dan analisis kustom. Pemrosesan khusus sangat berguna saat Anda berkolaborasi dan menganalisis data di banyak akun.

Misalnya, grup keamanan informasi perusahaan mungkin ingin menganalisis data untuk deteksi intrusi waktu nyata atau perilaku anomali agar bisa melakukan audit akun di semua divisi di perusahaan dengan mengumpulkan log produksi gabungan mereka untuk pemrosesan pusat. Aliran real-time data peristiwa di seluruh akun tersebut dapat dirakit dan dikirim ke grup keamanan informasi, yang dapat menggunakan Kinesis Data Streams untuk melampirkan data ke sistem analitik keamanan yang ada.

> **Note**
>
> Grup log dan tujuan harus berada di AWS Wilayah yang sama. Namun, AWS sumber daya yang ditunjuk tujuan dapat ditemukan di Wilayah yang berbeda. Dalam contoh di bagian berikut, semua sumber daya khusus Wilayah dibuat di AS Timur (Virginia N.)).

Topik

Berbagi data log lintas wilayah lintas akun menggunakan Kinesis Data Streams

Saat membuat langganan lintas akun, Anda dapat menentukan satu akun atau organisasi untuk menjadi pengirim. Jika Anda menentukan organisasi, maka prosedur ini memungkinkan semua akun di organisasi untuk mengirim log ke akun penerima.

Untuk berbagi data log lintas akun, Anda perlu membuat pengirim dan penerima data log:

- Pengirim data log —mendapatkan informasi tujuan dari penerima dan memberi tahu CloudWatch Log bahwa Log siap mengirim peristiwa lognya ke tujuan yang ditentukan. Dalam prosedur di bagian lainnya, pengirim data log ditampilkan dengan nomor AWS akun fiksi 111111111111.

  Jika Anda akan memiliki beberapa akun dalam satu organisasi yang mengirim log ke satu akun penerima, Anda dapat membuat kebijakan yang memberikan izin kepada semua akun di organisasi untuk mengirim log ke akun penerima. Anda masih harus menyiapkan filter langganan terpisah untuk setiap akun pengirim.
- Penerima data log —menyiapkan tujuan yang merangkum aliran Kinesis Data Streams dan CloudWatch memberi tahu Log bahwa penerima ingin menerima data log. Penerima kemudian membagikan informasi tentang tujuan ini dengan pengirim. Dalam prosedur di bagian lainnya, penerima data log ditampilkan dengan nomor AWS akun fiksi 999999999999.

Untuk mulai menerima peristiwa log dari pengguna lintas akun, penerima data log terlebih dahulu membuat tujuan CloudWatch Log. Setiap tujuan terdiri atas elemen kunci berikut:

Nama tujuan

Nama tujuan yang ingin Anda buat.

ARN Target

Nama Sumber Daya Amazon (ARN) dari AWS sumber daya yang ingin Anda gunakan sebagai tujuan umpan berlangganan.

ARN Peran

Peran AWS Identity and Access Management (IAM) yang memberikan CloudWatch Log izin yang diperlukan untuk memasukkan data ke aliran yang dipilih.

Kebijakan akses

Dokumen kebijakan IAM (dalam format JSON, ditulis menggunakan tata bahasa kebijakan IAM) yang mengatur set pengguna yang diizinkan untuk menulis ke tujuan Anda.

> **Note**
>
> Grup log dan tujuan harus berada di AWS Wilayah yang sama. Namun, sumber daya AWS yang ditunjuk oleh tujuan dapat berada di Wilayah yang berbeda. Dalam contoh di bagian berikut, semua sumber daya khusus Wilayah dibuat di US East (N. Virginia).

Topik

Menyiapkan langganan lintas akun baru

Ikuti langkah-langkah di bagian ini untuk menyiapkan langganan log lintas akun baru.

Topik

Langkah 1: Buat tujuan

> **Important**
>
> Semua langkah dalam prosedur ini harus dilakukan di akun penerima data log.

Untuk contoh ini, akun penerima data log memiliki ID akun 999999999999, sedangkan ID AWS akun pengirim AWS data log adalah 111111111111.

Contoh ini membuat tujuan menggunakan aliran Kinesis Data RecipientStream Streams yang disebut, dan peran CloudWatch yang memungkinkan Log untuk menulis data ke sana.

Saat tujuan dibuat, CloudWatch Log mengirimkan pesan pengujian ke tujuan atas nama akun penerima. Saat filter langganan aktif nanti, CloudWatch Log mengirimkan peristiwa log ke tujuan atas nama akun sumber.

Untuk membuat tujuan

1. Di akun penerima, buat aliran tujuan di Kinesis Data Streams. Di jendela perintah, ketik:

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Tunggu hingga streaming menjadi aktif. Anda dapat menggunakan perintah aws kinesis describe-stream untuk memeriksa. StreamDescription StreamStatusproperti. Selain itu, perhatikan StreamDescriptionnilai.streaMarn karena Anda akan meneruskannya ke CloudWatch Log nanti:

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

Mungkin diperlukan satu atau dua menit bagi pengaliran Anda untuk muncul dalam keadaan aktif.

3. Buat peran IAM yang memberikan izin kepada CloudWatch Log untuk memasukkan data ke aliran Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file ~/ TrustPolicyFor CWL.json. Gunakan editor teks untuk membuat file kebijakan ini, jangan menggunakan konsol IAM.

   Kebijakan ini mencakup kunci konteks kondisi `aws:SourceArn` global yang menentukan `sourceAccountId` untuk membantu mencegah masalah keamanan wakil yang membingungkan. Jika Anda belum mengetahui ID akun sumber pada panggilan pertama, kami

sarankan Anda memasukkan ARN tujuan di bidang ARN sumber. Dalam panggilan berikutnya, Anda harus mengatur ARN sumber menjadi ARN sumber sebenarnya yang Anda kumpulkan dari panggilan pertama. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. Gunakan perintah aws iam create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Perhatikan nilai Role.Arn yang dikembalikan karena itu juga akan diteruskan ke Log nanti: CloudWatch

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
```

```
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, gunakan editor teks untuk membuat kebijakan izin dalam file ~/ PermissionsFor CWL.json:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Kaitkan kebijakan izin dengan peran dengan menggunakan perintah aws iam: put-role-policy

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Setelah aliran dalam keadaan aktif dan Anda telah membuat peran IAM, Anda dapat membuat tujuan CloudWatch Log.

   a. Langkah ini tidak mengaitkan kebijakan akses dengan tujuan Anda dan hanya langkah pertama dari dua langkah yang menyelesaikan pembuatan tujuan. Catat DestinationArnyang dikembalikan dalam muatan:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

b. Setelah langkah 7a selesai, di akun penerima data log, kaitkan kebijakan akses dengan tujuan. Kebijakan ini harus menentukan PutSubscriptionFilter tindakan log: dan memberikan izin ke akun pengirim untuk mengakses tujuan.

   Kebijakan memberikan izin ke AWS akun yang mengirim log. Anda dapat menentukan hanya satu akun ini dalam kebijakan, atau jika akun pengirim adalah anggota organisasi, kebijakan dapat menentukan ID organisasi organisasi. Dengan cara ini, Anda dapat membuat hanya satu kebijakan untuk mengizinkan beberapa akun dalam satu organisasi mengirim log ke akun tujuan ini.

   Gunakan editor teks untuk membuat file bernama `~/AccessPolicy.json` dengan salah satu pernyataan kebijakan berikut.

   Kebijakan contoh pertama ini memungkinkan semua akun di organisasi yang memiliki ID `o-1234567890` untuk mengirim log ke akun penerima.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
```

```
                    "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

Contoh berikutnya ini memungkinkan hanya akun pengirim data log (111111111111) untuk mengirim log ke akun penerima data log.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. Lampirkan kebijakan yang Anda buat pada langkah sebelumnya ke tujuan.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Kebijakan akses ini memungkinkan pengguna di AWS Akun dengan ID 111111111111 untuk memanggil ***PutSubscriptionFilter****tujuan dengan ARN arn:aws:logs: region:999999999999:destination:testDestination.* Upaya pengguna lain untuk menelepon PutSubscriptionFilter terhadap tujuan ini akan ditolak.

Untuk memvalidasi hak istimewa pengguna berdasarkan kebijakan akses, lihat Menggunakan Validator Kebijakan dalam Panduan Pengguna IAM.

Setelah selesai, jika Anda menggunakan AWS Organizations izin lintas akun, ikuti langkah-langkahnya. Langkah 2: (Hanya jika menggunakan organisasi) Buat peran IAM Jika Anda memberikan izin langsung ke akun lain alih-alih menggunakan Organizations, Anda dapat melewati langkah itu dan melanjutkan ke. Langkah 4: Buat filter berlangganan

Langkah 2: (Hanya jika menggunakan organisasi) Buat peran IAM

Di bagian sebelumnya, jika Anda membuat tujuan menggunakan kebijakan akses yang memberikan izin kepada organisasi tempat akun `111111111111` berada, alih-alih memberikan izin langsung ke akun`111111111111`, ikuti langkah-langkah di bagian ini. Jika tidak, Anda dapat melompat keLangkah 4: Buat filter berlangganan.

Langkah-langkah di bagian ini membuat peran IAM, yang CloudWatch dapat mengasumsikan dan memvalidasi apakah akun pengirim memiliki izin untuk membuat filter langganan terhadap tujuan penerima.

Lakukan langkah-langkah di bagian ini di akun pengirim. Peran harus ada di akun pengirim, dan Anda menentukan ARN peran ini dalam filter langganan. Dalam contoh ini, akun pengirim adalah`111111111111`.

Untuk membuat peran IAM yang diperlukan untuk langganan log lintas akun menggunakan AWS Organizations

1. Buat kebijakan kepercayaan berikut dalam sebuah file`/TrustPolicyForCWLSubscriptionFilter.json`. Gunakan editor teks untuk membuat file kebijakan ini; jangan gunakan konsol IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Buat peran IAM yang menggunakan kebijakan ini. Perhatikan `Arn` nilai yang dikembalikan oleh perintah, Anda akan membutuhkannya nanti dalam prosedur ini. Dalam contoh ini, kita gunakan `CWLtoSubscriptionFilterRole` untuk nama peran yang kita buat.

```
aws iam create-role \
```

```
    --role-name CWLtoSubscriptionFilterRole \
    --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda.

   a. Pertama, gunakan editor teks untuk membuat kebijakan izin berikut dalam file bernama`~/PermissionsForCWLSubscriptionFilter.json`.

   ```
   {
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "logs:PutLogEvents",
               "Resource": "arn:aws:logs:region:111111111111:log-
   group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
           }
       ]
   }
   ```

   b. Masukkan perintah berikut untuk mengaitkan kebijakan izin yang baru saja Anda buat dengan peran yang Anda buat di langkah 2.

   ```
   aws iam put-role-policy
       --role-name CWLtoSubscriptionFilterRole
       --policy-name Permissions-Policy-For-CWL-Subscription-filter
       --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
   ```

Setelah selesai, Anda dapat melanjutkan keLangkah 4: Buat filter berlangganan.

Langkah 3: Tambah/validasi izin IAM untuk tujuan lintas akun

Menurut logika evaluasi kebijakan AWS lintas akun, untuk mengakses sumber daya lintas akun (seperti aliran Kinesis atau Firehose yang digunakan sebagai tujuan filter langganan), Anda harus memiliki kebijakan berbasis identitas di akun pengirim yang menyediakan akses eksplisit ke sumber tujuan lintas akun. Untuk informasi selengkapnya tentang logika evaluasi kebijakan, lihat Logika evaluasi kebijakan lintas akun.

Anda dapat melampirkan kebijakan berbasis identitas ke peran IAM atau pengguna IAM yang Anda gunakan untuk membuat filter langganan. Kebijakan ini harus ada di akun pengiriman. Jika Anda

menggunakan peran Administrator untuk membuat filter langganan, Anda dapat melewati langkah ini dan melanjutkan keLangkah 4: Buat filter berlangganan.

Untuk menambah atau memvalidasi izin IAM yang diperlukan untuk lintas akun

1. Masukkan perintah berikut untuk memeriksa peran IAM atau pengguna IAM mana yang digunakan untuk menjalankan perintah AWS log.

   ```
   aws sts get-caller-identity
   ```

   Perintah tersebut mengembalikan output serupa dengan berikut ini:

   ```
   {
   "UserId": "User ID",
   "Account": "sending account id",
   "Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
   }
   ```

   Catat nilai yang diwakili oleh *RoleName*atau *UserName*.
2. AWS Management Console Masuk ke akun pengiriman dan cari kebijakan terlampir dengan peran IAM atau pengguna IAM yang dikembalikan dalam output perintah yang Anda masukkan pada langkah 1.
3. Verifikasi bahwa kebijakan yang dilampirkan pada peran ini atau pengguna memberikan izin eksplisit untuk memanggil sumber `logs:PutSubscriptionFilter` daya tujuan lintas akun. Contoh kebijakan berikut menunjukkan izin yang disarankan.

   Kebijakan berikut memberikan izin untuk membuat filter langganan pada sumber daya tujuan apa pun hanya dalam satu AWS akun, akun`123456789012`:

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "Allow subscription filters on any resource in one specific
    account",
               "Effect": "Allow",
               "Action": "logs:PutSubscriptionFilter",
               "Resource": [
                   "arn:aws:logs:*:*:log-group:*",
                   "arn:aws:logs:*:123456789012:destination:*"
   ```

```
            ]
        }
    ]
}
```

Kebijakan berikut memberikan izin untuk membuat filter langganan hanya pada sumber daya tujuan tertentu yang dinamai `sampleDestination` dalam satu AWS akun, akun`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

Langkah 4: Buat filter berlangganan

Setelah Anda membuat tujuan, akun penerima data log dapat berbagi ARN tujuan (arn:aws:logs:us-east-1:999999999999:destination:testDestination) dengan akun AWS lain sehingga mereka dapat mengirim log acara ke tujuan yang sama. Para pengguna akun pengirim ini kemudian membuat filter langganan pada grup log masing-masing berdasarkan tujuan ini. Filter langganan segera memulai aliran data log waktu nyata dari grup log yang dipilih ke tujuan yang ditentukan.

> **Note**
>
> Jika Anda memberikan izin untuk filter langganan ke seluruh organisasi, Anda harus menggunakan ARN dari peran IAM yang Anda buat. Langkah 2: (Hanya jika menggunakan organisasi) Buat peran IAM

Dalam contoh berikut, filter langganan dibuat di akun pengiriman. filter dikaitkan dengan grup log yang berisi AWS CloudTrail peristiwa sehingga setiap aktivitas yang dicatat yang dibuat oleh AWS kredensyal “Root” dikirimkan ke tujuan yang Anda buat sebelumnya. Tujuan itu merangkum aliran yang disebut "". RecipientStream

Langkah-langkah lainnya di bagian berikut mengasumsikan bahwa Anda telah mengikuti petunjuk dalam Mengirim CloudTrail Acara ke CloudWatch Log di Panduan AWS CloudTrail Pengguna dan membuat grup log yang berisi CloudTrail peristiwa Anda. Langkah-langkah ini mengasumsikan bahwa nama grup log ini adalah`CloudTrail/logs`.

Saat memasukkan perintah berikut, pastikan Anda masuk sebagai pengguna IAM atau menggunakan peran IAM yang Anda tambahkan kebijakan untuk, masuk. Langkah 3: Tambah/validasi izin IAM untuk tujuan lintas akun

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Grup log dan tujuan harus berada di AWS Wilayah yang sama. Namun, tujuan dapat menunjuk ke AWS sumber daya seperti aliran Kinesis Data Streams yang terletak di Wilayah yang berbeda.

Validasi alur peristiwa log

Setelah Anda membuat filter langganan, CloudWatch Log meneruskan semua peristiwa log masuk yang cocok dengan pola filter ke aliran yang dienkapsulasi dalam aliran tujuan yang disebut "". RecipientStream Pemilik tujuan dapat memverifikasi bahwa ini terjadi dengan menggunakan get-shard-iterator perintah aws kinesis untuk mengambil pecahan Kinesis Data Streams, dan menggunakan perintah aws kinesis get-records untuk mengambil beberapa catatan Kinesis Data Streams:

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

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
```

```
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Anda mungkin perlu menjalankan kembali perintah get-records beberapa kali sebelum Kinesis Data Streams mulai mengembalikan data.

Anda akan melihat respons dengan array catatan Kinesis Data Streams. Atribut data dalam catatan Kinesis Data Streams dikompresi dalam format gzip dan kemudian base64 dikodekan. Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Data yang didekode dan didekompresi base64 diformat sebagai JSON dengan struktur berikut:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
```

```
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Elemen kunci dalam struktur data ini adalah sebagai berikut:

owner

ID AWS Akun dari data log asal.

logGroup

Nama grup log dari data log asal.

logStream

Nama pengaliran log dari data log asal.

subscriptionFilters

Daftar nama filter langganan yang cocok dengan data log asal.

messageType

Pesan data menggunakan tipe “DATA_MESSAGE”. Terkadang CloudWatch Log dapat memancarkan catatan Kinesis Data Streams dengan tipe “CONTROL_MESSAGE”, terutama untuk memeriksa apakah tujuan dapat dijangkau.

logEvents

Data log yang sebenarnya, direpresentasikan sebagai array catatan log acara. Properti ID adalah pengenal unik untuk setiap log acara.

Ubah keanggotaan tujuan saat runtime

Anda mungkin mengalami situasi ketika Anda harus menambahkan atau menghapus keanggotaan beberapa pengguna dari tujuan yang Anda miliki. Anda dapat menggunakan perintah `put-destination-policy` di tujuan Anda dengan kebijakan akses baru. Dalam contoh berikut, akun 111111111111 yang ditambahkan sudah sebelumnya dihentikan dari mengirim data log lagi, dan akun 222222222222 diaktifkan.

1. Ambil kebijakan yang saat ini terkait dengan TestDestination tujuan dan catat: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Perbarui kebijakan agar menunjukkan bahwa akun 111111111111 dihentikan, dan akun 222222222222 diaktifkan. Letakkan kebijakan ini di file ~/ NewAccessPolicy .json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
```

```
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Panggilan PutDestinationPolicyuntuk mengaitkan kebijakan yang ditentukan dalam NewAccessPolicyfile.json dengan tujuan:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Ini pada akhirnya akan menonaktifkan log acara dari ID akun 111111111111. Log acara dari ID akun 222222222222 mulai mengalir ke tujuan segera setelah pemilik akun 222222222222 membuat filter langganan.

Memperbarui langganan lintas akun yang ada

Jika saat ini Anda memiliki langganan log lintas akun di mana akun tujuan hanya memberikan izin ke akun pengirim tertentu, dan Anda ingin memperbarui langganan ini sehingga akun tujuan memberikan akses ke semua akun di organisasi, ikuti langkah-langkah di bagian ini.

Topik

- Langkah 1: Perbarui filter berlangganan
- Langkah 2: Perbarui kebijakan akses tujuan yang ada

Langkah 1: Perbarui filter berlangganan

> **Note**
>
> Langkah ini diperlukan hanya untuk langganan lintas akun untuk log yang dibuat oleh layanan yang tercantum di. Aktifkan pencatatan dari AWS layanan Jika Anda tidak bekerja dengan log yang dibuat oleh salah satu grup log ini, Anda dapat melompat ke Langkah 2: Perbarui kebijakan akses tujuan yang ada.

Dalam kasus tertentu, Anda harus memperbarui filter langganan di semua akun pengirim yang mengirim log ke akun tujuan. Pembaruan menambahkan peran IAM, yang CloudWatch dapat

mengasumsikan dan memvalidasi bahwa akun pengirim memiliki izin untuk mengirim log ke akun penerima.

Ikuti langkah-langkah di bagian ini untuk setiap akun pengirim yang ingin Anda perbarui untuk menggunakan ID organisasi untuk izin berlangganan lintas akun.

Dalam contoh di bagian ini, dua akun, 111111111111 dan 222222222222 sudah memiliki filter berlangganan yang dibuat untuk mengirim log ke akun999999999999. Nilai filter langganan yang ada adalah sebagai berikut:

```
## Existing Subscription Filter parameter values
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Jika Anda perlu menemukan nilai parameter filter langganan saat ini, masukkan perintah berikut.

```
aws logs describe-subscription-filters
    \ --log-group-name "my-log-group-name"
```

Untuk memperbarui filter langganan agar mulai menggunakan ID organisasi untuk izin log lintas akun

1. Buat kebijakan kepercayaan berikut dalam sebuah file~/TrustPolicyForCWL.json. Gunakan editor teks untuk membuat file kebijakan ini; jangan gunakan konsol IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Buat peran IAM yang menggunakan kebijakan ini. Perhatikan nilai Arn Arn nilai yang dikembalikan oleh perintah, Anda akan membutuhkannya nanti dalam prosedur ini. Dalam contoh ini, kita gunakan CWLtoSubscriptionFilterRole untuk nama peran yang kita buat.

   ```
   aws iam create-role
       \ --role-name CWLtoSubscriptionFilterRole
   ```

```
    \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

3. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda.

   a. Pertama, gunakan editor teks untuk membuat kebijakan izin berikut dalam file bernama`/PermissionsForCWLSubscriptionFilter.json`.

   ```
   {
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "logs:PutLogEvents",
               "Resource": "arn:aws:logs:region:111111111111:log-
   group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
           }
       ]
   }
   ```

   b. Masukkan perintah berikut untuk mengaitkan kebijakan izin yang baru saja Anda buat dengan peran yang Anda buat di langkah 2.

   ```
   aws iam put-role-policy
       --role-name CWLtoSubscriptionFilterRole
       --policy-name Permissions-Policy-For-CWL-Subscription-filter
       --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
   ```

4. Masukkan perintah berikut untuk memperbarui filter langganan.

```
aws logs put-subscription-filter
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn
 "arn:aws:logs:region:999999999999:destination:testDestination"
    \ --role-arn "arn:aws:iam::111111111111:role/CWLtoSubscriptionFilterRole"
```

Langkah 2: Perbarui kebijakan akses tujuan yang ada

Setelah memperbarui filter langganan di semua akun pengirim, Anda dapat memperbarui kebijakan akses tujuan di akun penerima.

Dalam contoh berikut, akun penerima adalah 999999999999 dan tujuan diberi nama`testDestination`.

Pembaruan memungkinkan semua akun yang merupakan bagian dari organisasi dengan ID `o-1234567890` untuk mengirim log ke akun penerima. Hanya akun yang memiliki filter langganan yang dibuat yang benar-benar akan mengirim log ke akun penerima.

Untuk memperbarui kebijakan akses tujuan di akun penerima untuk mulai menggunakan ID organisasi untuk izin

1. Di akun penerima, gunakan editor teks untuk membuat `~/AccessPolicy.json` file dengan konten berikut.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

2. Masukkan perintah berikut untuk melampirkan kebijakan yang baru saja Anda buat ke tujuan yang ada. Untuk memperbarui tujuan agar menggunakan kebijakan akses dengan ID organisasi, bukan kebijakan akses yang mencantumkan ID AWS akun tertentu, sertakan `force` parameternya.

> **Warning**
>
> Jika Anda bekerja dengan log yang dikirim oleh AWS layanan yang terdaftar di[Aktifkan pencatatan dari AWS layanan](), maka sebelum melakukan langkah ini, Anda harus

terlebih dahulu memperbarui filter langganan di semua akun pengirim seperti yang dijelaskan di[Langkah 1: Perbarui filter berlangganan](#).

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Berbagi data log lintas wilayah lintas akun menggunakan Firehose

Untuk berbagi data log lintas akun, Anda perlu membuat pengirim dan penerima data log:

- Pengirim data log —mendapatkan informasi tujuan dari penerima dan memberi tahu CloudWatch Log bahwa ia siap untuk mengirim peristiwa lognya ke tujuan yang ditentukan. Dalam prosedur di bagian lainnya, pengirim data log ditampilkan dengan nomor AWS akun fiksi 111111111111.
- Penerima data log —menyiapkan tujuan yang merangkum aliran Kinesis Data Streams dan CloudWatch memberi tahu Log bahwa penerima ingin menerima data log. Penerima kemudian membagikan informasi tentang tujuan ini dengan pengirim. Dalam prosedur di bagian lainnya, penerima data log ditampilkan dengan nomor AWS akun fiksi 22222222222222.

Contoh di bagian ini menggunakan aliran pengiriman Firehose dengan penyimpanan Amazon S3. Anda juga dapat mengatur aliran pengiriman Firehose dengan pengaturan berbeda. Untuk informasi selengkapnya, lihat Membuat Aliran Pengiriman Firehose.

> **Note**
>
> Grup log dan tujuan harus berada di AWS Wilayah yang sama. Namun, sumber daya AWS yang ditunjuk oleh tujuan dapat berada di Wilayah yang berbeda.

> **Note**
>
> Filter langganan Firehose untuk akun yang sama dan aliran pengiriman lintas wilayah didukung.

Topik

Langkah 1: Buat aliran pengiriman Firehose

> **Important**
>
> Sebelum Anda menyelesaikan langkah-langkah berikut, Anda harus menggunakan kebijakan akses, sehingga Firehose dapat mengakses bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat Mengontrol Akses di Panduan Pengembang Amazon Data Firehose.
> Semua langkah di bagian ini (Langkah 1) harus dilakukan di akun penerima data log.
> US East (N. Virginia) digunakan dalam contoh perintah berikut. Ganti Wilayah ini dengan Wilayah yang benar untuk penerapan Anda.

Untuk membuat aliran pengiriman Firehose yang akan digunakan sebagai tujuan

1. Buat bucket Amazon S3:

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. Buat peran IAM yang memberikan izin Firehose untuk memasukkan data ke dalam bucket.

   a. Pertama, gunakan editor teks untuk membuat kebijakan kepercayaan dalam file `~/TrustPolicyForFirehose.json`.

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

   b. Buat IAM role dengan menentukan file kebijakan kepercayaan yang baru saja Anda buat.

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

c. Output perintah ini akan terlihat serupa dengan yang berikut ini. Catat nama peran dan ARN peran.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan Firehose di akun Anda.

a. Pertama, gunakan editor teks untuk membuat kebijakan izin berikut dalam file bernama`~/PermissionsForFirehose.json`. Bergantung pada kasus penggunaan Anda, Anda mungkin perlu menambahkan lebih banyak izin ke file ini.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
```

```
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
    }]
}
```

b. Masukkan perintah berikut untuk mengaitkan kebijakan izin yang baru saja Anda buat dengan peran IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Masukkan perintah berikut untuk membuat aliran pengiriman Firehose. Ganti *my-role-arn* dan *my-bucket-arn* dengan nilai yang benar untuk penerapan Anda.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

Outputnya akan serupa dengan yang berikut ini:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Langkah 2: Buat tujuan

> **Important**
>
> Semua langkah dalam prosedur ini harus dilakukan di akun penerima data log.

Saat tujuan dibuat, CloudWatch Log mengirimkan pesan pengujian ke tujuan atas nama akun penerima. Saat filter langganan aktif nanti, CloudWatch Log mengirimkan peristiwa log ke tujuan atas nama akun sumber.

Untuk membuat tujuan

1. Tunggu hingga aliran Firehose yang Anda buat Langkah 1: Buat aliran pengiriman Firehose menjadi aktif. Anda dapat menggunakan perintah berikut untuk memeriksa StreamDescription. StreamStatusproperti.

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
```

   Selain itu, perhatikan DeliveryStreamDescription. DeliveryStreamNilai ARN, karena Anda harus menggunakannya di langkah selanjutnya. Contoh output dari perintah ini:

```
{
    "DeliveryStreamDescription": {
        "DeliveryStreamName": "my-delivery-stream",
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:222222222222:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamEncryptionConfiguration": {
            "Status": "DISABLED"
        },
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
```

```
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Mungkin diperlukan satu atau dua menit bagi aliran pengiriman Anda untuk muncul dalam keadaan aktif.

2. Saat aliran pengiriman aktif, buat peran IAM yang akan memberikan izin kepada CloudWatch Log untuk memasukkan data ke aliran Firehose Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file ~/ TrustPolicyFor CWL.json. Gunakan editor teks untuk membuat kebijakan ini. Untuk informasi selengkapnya tentang titik akhir CloudWatch Log, lihat titik akhir dan CloudWatch kuota Amazon Logs.

   Kebijakan ini mencakup kunci konteks kondisi `aws:SourceArn` global yang menentukan `sourceAccountId` untuk membantu mencegah masalah keamanan wakil yang membingungkan. Jika Anda belum mengetahui ID akun sumber pada panggilan pertama, kami sarankan Anda memasukkan ARN tujuan di bidang ARN sumber. Dalam panggilan berikutnya, Anda harus mengatur ARN sumber menjadi ARN sumber sebenarnya yang Anda kumpulkan dari panggilan pertama. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

   ```
   {
   ```

```
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.region.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
     }
}
```

3. Gunakan perintah aws iam create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan yang baru saja Anda buat.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

Berikut ini adalah contoh output. Perhatikan nilai `Role.Arn` yang dikembalikan, karena Anda akan perlu menggunakannya di langkah berikutnya.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2021-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
```

```
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, gunakan editor teks untuk membuat kebijakan izin dalam file ~/ PermissionsFor CWL.json:

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Kaitkan kebijakan izin dengan peran tersebut dengan memasukkan perintah berikut:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Setelah aliran pengiriman Firehose dalam status aktif dan Anda telah membuat peran IAM, Anda dapat membuat tujuan Log. CloudWatch

   a. Langkah ini tidak akan mengaitkan kebijakan akses dengan tujuan Anda dan hanya merupakan langkah pertama dari dua langkah yang akan menyelesaikan pembuatan tujuan. Catat ARN tujuan baru yang dikembalikan di payload, karena Anda akan menggunakan ini sebagai langkah `destination.arn` selanjutnya.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
```

```
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

b. Setelah langkah sebelumnya selesai, dalam akun penerima data log (222222222222), kaitkan kebijakan akses dengan tujuan.

   Kebijakan ini memungkinkan akun pengirim data log (11111111111111) untuk mengakses tujuan hanya di akun penerima data log (222222222222). Anda dapat menggunakan editor teks untuk meletakkan kebijakan ini di file ~/ AccessPolicy .json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. Ini membuat kebijakan yang menentukan siapa yang memiliki akses menulis ke tujuan. Kebijakan ini harus menentukan PutSubscriptionFilter tindakan log: untuk mengakses tujuan. Pengguna lintas akun akan menggunakan PutSubscriptionFiltertindakan untuk mengirim peristiwa log ke tujuan:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Langkah 3: Tambah/validasi izin IAM untuk tujuan lintas akun

Menurut logika evaluasi kebijakan AWS lintas akun, untuk mengakses sumber daya lintas akun (seperti aliran Kinesis atau Firehose yang digunakan sebagai tujuan filter langganan), Anda harus memiliki kebijakan berbasis identitas di akun pengirim yang menyediakan akses eksplisit ke sumber tujuan lintas akun. Untuk informasi selengkapnya tentang logika evaluasi kebijakan, lihat Logika evaluasi kebijakan lintas akun.

Anda dapat melampirkan kebijakan berbasis identitas ke peran IAM atau pengguna IAM yang Anda gunakan untuk membuat filter langganan. Kebijakan ini harus ada di akun pengiriman. Jika Anda menggunakan peran Administrator untuk membuat filter langganan, Anda dapat melewati langkah ini dan melanjutkan keLangkah 4: Buat filter berlangganan.

Untuk menambah atau memvalidasi izin IAM yang diperlukan untuk lintas akun

1. Masukkan perintah berikut untuk memeriksa peran IAM atau pengguna IAM mana yang digunakan untuk menjalankan perintah AWS log.

   ```
   aws sts get-caller-identity
   ```

   Perintah tersebut mengembalikan output serupa dengan berikut ini:

   ```
   {
   "UserId": "User ID",
   "Account": "sending account id",
   "Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
   }
   ```

   Catat nilai yang diwakili oleh *RoleName*atau *UserName*.

2. AWS Management Console Masuk ke akun pengiriman dan cari kebijakan terlampir dengan peran IAM atau pengguna IAM yang dikembalikan dalam output perintah yang Anda masukkan pada langkah 1.

3. Verifikasi bahwa kebijakan yang dilampirkan pada peran ini atau pengguna memberikan izin eksplisit untuk memanggil sumber `logs:PutSubscriptionFilter` daya tujuan lintas akun. Contoh kebijakan berikut menunjukkan izin yang disarankan.

   Kebijakan berikut memberikan izin untuk membuat filter langganan pada sumber daya tujuan apa pun hanya dalam satu AWS akun, akun`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on any resource in one specific
 account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

   Kebijakan berikut memberikan izin untuk membuat filter langganan hanya pada sumber daya tujuan tertentu yang dinamai `sampleDestination` dalam satu AWS akun, akun`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
```

```
}
```

Langkah 4: Buat filter berlangganan

Beralihlah ke akun pengiriman, yaitu 111111111111 dalam contoh ini. Sekarang Anda akan membuat filter langganan di akun pengirim. Dalam contoh ini, filter dikaitkan dengan grup log yang berisi AWS CloudTrail peristiwa sehingga setiap aktivitas yang dicatat yang dibuat oleh AWS kredensyal "Root" dikirimkan ke tujuan yang sebelumnya Anda buat. Untuk informasi selengkapnya tentang cara mengirim AWS CloudTrail peristiwa ke CloudWatch Log, lihat Mengirim CloudTrail Acara ke CloudWatch Log di Panduan AWS CloudTrail Pengguna.

Saat memasukkan perintah berikut, pastikan Anda masuk sebagai pengguna IAM atau menggunakan peran IAM yang Anda tambahkan kebijakan untuk, masuk. Langkah 3: Tambah/validasi izin IAM untuk tujuan lintas akun

```
aws logs put-subscription-filter \
    --log-group-name "aws-cloudtrail-logs-111111111111-300a971e" \
    --filter-name "firehose_test" \
    --filter-pattern "{$.userIdentity.type = AssumedRole}" \
    --destination-arn "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
```

Grup log dan tujuan harus berada di AWS Wilayah yang sama. Namun, tujuan dapat menunjuk ke AWS sumber daya seperti aliran Firehose yang terletak di Wilayah yang berbeda.

Memvalidasi alur peristiwa log

Setelah Anda membuat filter langganan, CloudWatch Log meneruskan semua peristiwa log masuk yang cocok dengan pola filter ke aliran pengiriman Firehose. Data mulai muncul di bucket Amazon S3 Anda berdasarkan interval buffer waktu yang disetel pada aliran pengiriman Firehose. Setelah waktu tertentu berlalu, Anda dapat memverifikasi data dengan memeriksa bucket Amazon S3. Untuk memeriksa bucket, masukkan perintah berikut:

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

Output perintah tersebut akan serupa dengan yang berikut ini:

```
{
    "Contents": [
```

```
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2021-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

Anda kemudian dapat mengambil objek tertentu dari bucket dengan memasukkan perintah berikut. Ganti nilai `key` dengan nilai yang Anda temukan di perintah sebelumnya.

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Data dalam objek Amazon S3 dikompresi dengan format gzip. Anda dapat memeriksa data mentah dari baris perintah menggunakan salah satu dari perintah berikut:

Linux:

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Memodifikasi keanggotaan tujuan saat runtime

Anda mungkin mengalami situasi ketika Anda harus menambahkan atau menghapus pengirim log dari tujuan yang Anda miliki. Anda dapat menggunakan PutDestinationPolicytindakan di tujuan Anda dengan kebijakan akses baru. Dalam contoh berikut, akun 111111111111 yang ditambahkan sudah sebelumnya dihentikan dari mengirim data log lagi, dan akun 333333333333 diaktifkan.

1. Ambil kebijakan yang saat ini terkait dengan TestDestination tujuan dan catat: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testFirehoseDestination"

{
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Perbarui kebijakan agar menunjukkan bahwa akun 111111111111 dihentikan, dan akun 333333333333 diaktifkan. Letakkan kebijakan ini di file ~/ NewAccessPolicy .json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Gunakan perintah berikut untuk mengaitkan kebijakan yang ditentukan dalam NewAccessPolicyfile.json dengan tujuan:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

   Ini akhirnya akan menonaktifkan log acara dari ID akun 111111111111. Log acara dari ID akun 333333333333 mulai mengalir ke tujuan segera setelah pemilik akun 333333333333 membuat filter langganan.

Langganan tingkat akun lintas wilayah lintas akun menggunakan Kinesis Data Streams

Saat membuat langganan lintas akun, Anda dapat menentukan satu akun atau organisasi untuk menjadi pengirim. Jika Anda menentukan organisasi, maka prosedur ini memungkinkan semua akun di organisasi untuk mengirim log ke akun penerima.

Untuk berbagi data log lintas akun, Anda perlu membuat pengirim dan penerima data log:

- Pengirim data log —mendapatkan informasi tujuan dari penerima dan memberi tahu CloudWatch Log bahwa Log siap mengirim peristiwa lognya ke tujuan yang ditentukan. Dalam prosedur di bagian lainnya, pengirim data log ditampilkan dengan nomor AWS akun fiksi 111111111111.

  Jika Anda akan memiliki beberapa akun dalam satu organisasi yang mengirim log ke satu akun penerima, Anda dapat membuat kebijakan yang memberikan izin kepada semua akun di organisasi untuk mengirim log ke akun penerima. Anda masih harus menyiapkan filter langganan terpisah untuk setiap akun pengirim.
- Penerima data log —menyiapkan tujuan yang merangkum aliran Kinesis Data Streams dan CloudWatch memberi tahu Log bahwa penerima ingin menerima data log. Penerima kemudian membagikan informasi tentang tujuan ini dengan pengirim. Dalam prosedur di bagian lainnya, penerima data log ditampilkan dengan nomor AWS akun fiksi 999999999999.

Untuk mulai menerima peristiwa log dari pengguna lintas akun, penerima data log terlebih dahulu membuat tujuan CloudWatch Log. Setiap tujuan terdiri atas elemen kunci berikut:

Nama tujuan

Nama tujuan yang ingin Anda buat.

ARN Target

Nama Sumber Daya Amazon (ARN) dari AWS sumber daya yang ingin Anda gunakan sebagai tujuan umpan berlangganan.

ARN Peran

Peran AWS Identity and Access Management (IAM) yang memberikan CloudWatch Log izin yang diperlukan untuk memasukkan data ke aliran yang dipilih.

Kebijakan akses

Dokumen kebijakan IAM (dalam format JSON, ditulis menggunakan tata bahasa kebijakan IAM) yang mengatur set pengguna yang diizinkan untuk menulis ke tujuan Anda.

> **Note**
>
> Grup log dan tujuan harus berada di AWS Wilayah yang sama. Namun, sumber daya AWS yang ditunjuk oleh tujuan dapat berada di Wilayah yang berbeda. Dalam contoh di bagian berikut, semua sumber daya khusus Wilayah dibuat di US East (N. Virginia).

Topik

Menyiapkan langganan lintas akun baru

Ikuti langkah-langkah di bagian ini untuk menyiapkan langganan log lintas akun baru.

Topik

- Ubah keanggotaan tujuan saat runtime

Langkah 1: Buat tujuan

> **Important**
>
> Semua langkah dalam prosedur ini harus dilakukan di akun penerima data log.

Untuk contoh ini, akun penerima data log memiliki ID akun 999999999999, sedangkan ID AWS akun pengirim AWS data log adalah 111111111111.

Contoh ini membuat tujuan menggunakan aliran Kinesis Data RecipientStream Streams yang disebut, dan peran CloudWatch yang memungkinkan Log untuk menulis data ke sana.

Saat tujuan dibuat, CloudWatch Log mengirimkan pesan pengujian ke tujuan atas nama akun penerima. Saat filter langganan aktif nanti, CloudWatch Log mengirimkan peristiwa log ke tujuan atas nama akun sumber.

Untuk membuat tujuan

1. Di akun penerima, buat aliran tujuan di Kinesis Data Streams. Di jendela perintah, ketik:

   ```
   aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
   ```

2. Tunggu hingga streaming menjadi aktif. Anda dapat menggunakan perintah aws kinesis describe-stream untuk memeriksa. StreamDescription StreamStatusproperti. Selain itu, perhatikan StreamDescriptionnilai.streaMarn karena Anda akan meneruskannya ke CloudWatch Log nanti:

   ```
   aws kinesis describe-stream --stream-name "RecipientStream"
   {
     "StreamDescription": {
       "StreamStatus": "ACTIVE",
       "StreamName": "RecipientStream",
       "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
       "Shards": [
         {
           "ShardId": "shardId-000000000000",
           "HashKeyRange": {
   ```

```
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702l8EXAMPLE"
        }
      }
    ]
  }
}
```

Mungkin diperlukan satu atau dua menit bagi pengaliran Anda untuk muncul dalam keadaan aktif.

3. Buat peran IAM yang memberikan izin kepada CloudWatch Log untuk memasukkan data ke aliran Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file ~/ TrustPolicyFor CWL.json. Gunakan editor teks untuk membuat file kebijakan ini, jangan menggunakan konsol IAM.

   Kebijakan ini mencakup kunci konteks kondisi `aws:SourceArn` global yang menentukan `sourceAccountId` untuk membantu mencegah masalah keamanan wakil yang membingungkan. Jika Anda belum mengetahui ID akun sumber pada panggilan pertama, kami sarankan Anda memasukkan ARN tujuan di bidang ARN sumber. Dalam panggilan berikutnya, Anda harus mengatur ARN sumber menjadi ARN sumber sebenarnya yang Anda kumpulkan dari panggilan pertama. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
```

```
    }
}
```

4. Gunakan perintah aws iam create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan. Perhatikan nilai Role.Arn yang dikembalikan karena itu juga akan diteruskan ke Log nanti: CloudWatch

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, gunakan editor teks untuk membuat kebijakan izin dalam file ~/ PermissionsFor CWL.json:

```
{
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Kaitkan kebijakan izin dengan peran dengan menggunakan perintah aws iam: put-role-policy

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Setelah aliran dalam keadaan aktif dan Anda telah membuat peran IAM, Anda dapat membuat tujuan CloudWatch Log.

   a. Langkah ini tidak mengaitkan kebijakan akses dengan tujuan Anda dan hanya langkah pertama dari dua langkah yang menyelesaikan pembuatan tujuan. Catat DestinationArnyang dikembalikan dalam muatan:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Setelah langkah 7a selesai, di akun penerima data log, kaitkan kebijakan akses dengan tujuan. Kebijakan ini harus menentukan PutSubscriptionFilter tindakan log: dan memberikan izin ke akun pengirim untuk mengakses tujuan.

      Kebijakan memberikan izin ke AWS akun yang mengirim log. Anda dapat menentukan hanya satu akun ini dalam kebijakan, atau jika akun pengirim adalah anggota organisasi,

kebijakan dapat menentukan ID organisasi organisasi. Dengan cara ini, Anda dapat membuat hanya satu kebijakan untuk mengizinkan beberapa akun dalam satu organisasi mengirim log ke akun tujuan ini.

Gunakan editor teks untuk membuat file bernama `~/AccessPolicy.json` dengan salah satu pernyataan kebijakan berikut.

Kebijakan contoh pertama ini memungkinkan semua akun di organisasi yang memiliki ID `o-1234567890` untuk mengirim log ke akun penerima.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

Contoh berikutnya ini memungkinkan hanya akun pengirim data log (111111111111) untuk mengirim log ke akun penerima data log.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
```

```
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. Lampirkan kebijakan yang Anda buat pada langkah sebelumnya ke tujuan.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Kebijakan akses ini memungkinkan pengguna di AWS Akun dengan ID 111111111111 untuk memanggil ***PutSubscriptionFilter****tujuan dengan ARN arn:aws:logs: region:999999999999:destination:testDestination.* Upaya pengguna lain untuk menelepon PutSubscriptionFilter terhadap tujuan ini akan ditolak.

Untuk memvalidasi hak istimewa pengguna berdasarkan kebijakan akses, lihat Menggunakan Validator Kebijakan dalam Panduan Pengguna IAM.

Setelah selesai, jika Anda menggunakan AWS Organizations izin lintas akun, ikuti langkah-langkahnya. Langkah 2: (Hanya jika menggunakan organisasi) Buat peran IAM Jika Anda memberikan izin langsung ke akun lain alih-alih menggunakan Organizations, Anda dapat melewati langkah itu dan melanjutkan ke. Langkah 3: Buat kebijakan filter langganan tingkat akun

Langkah 2: (Hanya jika menggunakan organisasi) Buat peran IAM

Di bagian sebelumnya, jika Anda membuat tujuan menggunakan kebijakan akses yang memberikan izin kepada organisasi tempat akun 111111111111 berada, alih-alih memberikan izin langsung ke akun111111111111, ikuti langkah-langkah di bagian ini. Jika tidak, Anda dapat melompat keLangkah 3: Buat kebijakan filter langganan tingkat akun.

Langkah-langkah di bagian ini membuat peran IAM, yang CloudWatch dapat mengasumsikan dan memvalidasi apakah akun pengirim memiliki izin untuk membuat filter langganan terhadap tujuan penerima.

Lakukan langkah-langkah di bagian ini di akun pengirim. Peran harus ada di akun pengirim, dan Anda menentukan ARN peran ini dalam filter langganan. Dalam contoh ini, akun pengirim adalah111111111111.

Untuk membuat peran IAM yang diperlukan untuk langganan log lintas akun menggunakan AWS Organizations

1. Buat kebijakan kepercayaan berikut dalam sebuah file`/TrustPolicyForCWLSubscriptionFilter.json`. Gunakan editor teks untuk membuat file kebijakan ini; jangan gunakan konsol IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Buat peran IAM yang menggunakan kebijakan ini. Perhatikan `Arn` nilai yang dikembalikan oleh perintah, Anda akan membutuhkannya nanti dalam prosedur ini. Dalam contoh ini, kita gunakan `CWLtoSubscriptionFilterRole` untuk nama peran yang kita buat.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda.

   a. Pertama, gunakan editor teks untuk membuat kebijakan izin berikut dalam file bernama`~/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Masukkan perintah berikut untuk mengaitkan kebijakan izin yang baru saja Anda buat dengan peran yang Anda buat di langkah 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Setelah selesai, Anda dapat melanjutkan keLangkah 3: Buat kebijakan filter langganan tingkat akun.

Langkah 3: Buat kebijakan filter langganan tingkat akun

Setelah Anda membuat tujuan, akun penerima data log dapat berbagi ARN tujuan (arn:aws:logs:us-east-1:999999999999:destination:testDestination) dengan akun AWS lain sehingga mereka dapat mengirim log acara ke tujuan yang sama. Para pengguna akun pengirim ini kemudian membuat filter langganan pada grup log masing-masing berdasarkan tujuan ini. Filter langganan segera memulai aliran data log waktu nyata dari grup log yang dipilih ke tujuan yang ditentukan.

> **Note**
>
> Jika Anda memberikan izin untuk filter langganan ke seluruh organisasi, Anda harus menggunakan ARN dari peran IAM yang Anda buat. Langkah 2: (Hanya jika menggunakan organisasi) Buat peran IAM

Dalam contoh berikut, kebijakan filter langganan tingkat akun dibuat di akun pengiriman. filter dikaitkan dengan akun pengirim 111111111111 sehingga setiap peristiwa log yang cocok dengan filter dan kriteria pemilihan dikirim ke tujuan yang Anda buat sebelumnya. Tujuan itu merangkum aliran yang disebut "". RecipientStream

`selection-criteria`Bidang ini opsional, tetapi penting untuk mengecualikan grup log yang dapat menyebabkan rekursi log tak terbatas dari filter langganan. Untuk informasi selengkapnya tentang masalah ini dan menentukan grup log mana yang akan dikecualikan, lihatPencegahan rekursi log. Saat ini, NOT IN adalah satu-satunya operator yang didukung untuk`selection-criteria`.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
```

```
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Grup log akun pengirim dan tujuan harus berada di AWS Wilayah yang sama. Namun, tujuan dapat menunjuk ke AWS sumber daya seperti aliran Kinesis Data Streams yang terletak di Wilayah yang berbeda.

Validasi alur peristiwa log

Setelah Anda membuat kebijakan filter langganan tingkat akun, CloudWatch Log meneruskan semua peristiwa log masuk yang cocok dengan pola filter dan kriteria pemilihan ke aliran yang dienkapsulasi dalam aliran tujuan yang disebut "". RecipientStream Pemilik tujuan dapat memverifikasi bahwa ini terjadi dengan menggunakan get-shard-iterator perintah aws kinesis untuk mengambil pecahan Kinesis Data Streams, dan menggunakan perintah aws kinesis get-records untuk mengambil beberapa catatan Kinesis Data Streams:

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

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Anda mungkin perlu menjalankan kembali `get-records` perintah beberapa kali sebelum Kinesis Data Streams mulai mengembalikan data.

Anda akan melihat respons dengan array catatan Kinesis Data Streams. Atribut data dalam catatan Kinesis Data Streams dikompresi dalam format gzip dan kemudian base64 dikodekan. Anda dapat memeriksa data mentah dari baris perintah menggunakan perintah Unix berikut:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Data yang didekode dan didekompresi base64 diformat sebagai JSON dengan struktur berikut:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
```

```
}
```

Elemen kunci dalam struktur data adalah sebagai berikut:

messageType

Pesan data akan menggunakan tipe "DATA_MESSAGE". Terkadang CloudWatch Log mungkin memancarkan catatan Kinesis Data Streams dengan tipe “CONTROL_MESSAGE”, terutama untuk memeriksa apakah tujuan dapat dijangkau.

owner

ID AWS Akun dari data log asal.

logGroup

Nama grup log dari data log asal.

logStream

Nama pengaliran log dari data log asal.

subscriptionFilters

Daftar nama filter langganan yang cocok dengan data log asal.

logEvents

Data log yang sebenarnya, direpresentasikan sebagai array catatan log acara. Properti "id" adalah pengenal unik untuk setiap log acara.

PolicyLevel

Tingkat di mana kebijakan itu ditegakkan. “ACCOUNT_LEVEL_POLICY” adalah `policyLevel` untuk kebijakan filter langganan tingkat akun.

Ubah keanggotaan tujuan saat runtime

Anda mungkin mengalami situasi ketika Anda harus menambahkan atau menghapus keanggotaan beberapa pengguna dari tujuan yang Anda miliki. Anda dapat menggunakan perintah `put-destination-policy` di tujuan Anda dengan kebijakan akses baru. Dalam contoh berikut, akun 111111111111 yang ditambahkan sudah sebelumnya dihentikan dari mengirim data log lagi, dan akun 222222222222 diaktifkan.

1. Ambil kebijakan yang saat ini terkait dengan TestDestination tujuan dan catat: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Perbarui kebijakan agar menunjukkan bahwa akun 111111111111 dihentikan, dan akun 222222222222 diaktifkan. Letakkan kebijakan ini di file ~/ NewAccessPolicy .json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Panggilan PutDestinationPolicyuntuk mengaitkan kebijakan yang ditentukan dalam NewAccessPolicyfile.json dengan tujuan:

```
aws logs put-destination-policy \
```

```
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Ini pada akhirnya akan menonaktifkan log acara dari ID akun 111111111111. Log acara dari ID akun 222222222222 mulai mengalir ke tujuan segera setelah pemilik akun 222222222222 membuat filter langganan.

Memperbarui langganan lintas akun yang ada

Jika saat ini Anda memiliki langganan log lintas akun di mana akun tujuan hanya memberikan izin ke akun pengirim tertentu, dan Anda ingin memperbarui langganan ini sehingga akun tujuan memberikan akses ke semua akun di organisasi, ikuti langkah-langkah di bagian ini.

Topik

Langkah 1: Perbarui filter berlangganan

> **Note**
>
> Langkah ini diperlukan hanya untuk langganan lintas akun untuk log yang dibuat oleh layanan yang tercantum di. Aktifkan pencatatan dari AWS layanan Jika Anda tidak bekerja dengan log yang dibuat oleh salah satu grup log ini, Anda dapat melompat ke Langkah 2: Perbarui kebijakan akses tujuan yang ada.

Dalam kasus tertentu, Anda harus memperbarui filter langganan di semua akun pengirim yang mengirim log ke akun tujuan. Pembaruan menambahkan peran IAM, yang CloudWatch dapat mengasumsikan dan memvalidasi bahwa akun pengirim memiliki izin untuk mengirim log ke akun penerima.

Ikuti langkah-langkah di bagian ini untuk setiap akun pengirim yang ingin Anda perbarui untuk menggunakan ID organisasi untuk izin berlangganan lintas akun.

Dalam contoh di bagian ini, dua akun, 111111111111 dan 222222222222 sudah memiliki filter berlangganan yang dibuat untuk mengirim log ke akun999999999999. Nilai filter langganan yang ada adalah sebagai berikut:

```
## Existing Subscription Filter parameter values
{
    "DestinationArn": "arn:aws:logs:region:999999999999:destination:testDestination",
    "FilterPattern": "{$.userIdentity.type = Root}",
    "Distribution": "Random"
}
```

Jika Anda perlu menemukan nilai parameter filter langganan saat ini, masukkan perintah berikut.

```
aws logs describe-account-policies \
--policy-type "SUBSCRIPTION_FILTER_POLICY" \
--policy-name "CrossAccountStreamsExamplePolicy"
```

Untuk memperbarui filter langganan agar mulai menggunakan ID organisasi untuk izin log lintas akun

1. Buat kebijakan kepercayaan berikut dalam sebuah file`~/TrustPolicyForCWL.json`. Gunakan editor teks untuk membuat file kebijakan ini; jangan gunakan konsol IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Buat peran IAM yang menggunakan kebijakan ini. Perhatikan nilai `Arn` `Arn` nilai yang dikembalikan oleh perintah, Anda akan membutuhkannya nanti dalam prosedur ini. Dalam contoh ini, kita gunakan `CWLtoSubscriptionFilterRole` untuk nama peran yang kita buat.

   ```
   aws iam create-role
       \ --role-name CWLtoSubscriptionFilterRole
       \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
   ```

3. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda.

   a. Pertama, gunakan editor teks untuk membuat kebijakan izin berikut dalam file bernama`~/PermissionsForCWLSubscriptionFilter.json`.

      ```
      {
      ```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Masukkan perintah berikut untuk mengaitkan kebijakan izin yang baru saja Anda buat dengan peran yang Anda buat di langkah 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. Masukkan perintah berikut untuk memperbarui kebijakan filter langganan.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "{$.userIdentity.type = Root}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Langkah 2: Perbarui kebijakan akses tujuan yang ada

Setelah memperbarui filter langganan di semua akun pengirim, Anda dapat memperbarui kebijakan akses tujuan di akun penerima.

Dalam contoh berikut, akun penerima adalah 999999999999 dan tujuan diberi nama`testDestination`.

Pembaruan memungkinkan semua akun yang merupakan bagian dari organisasi dengan ID `o-1234567890` untuk mengirim log ke akun penerima. Hanya akun yang memiliki filter langganan yang dibuat yang benar-benar akan mengirim log ke akun penerima.

Untuk memperbarui kebijakan akses tujuan di akun penerima untuk mulai menggunakan ID organisasi untuk izin

1. Di akun penerima, gunakan editor teks untuk membuat `~/AccessPolicy.json` file dengan konten berikut.

```
{
"Version" : "2012-10-17",
    "Statement" : [
        {
"Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
"StringEquals" : {
"aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

2. Masukkan perintah berikut untuk melampirkan kebijakan yang baru saja Anda buat ke tujuan yang ada. Untuk memperbarui tujuan agar menggunakan kebijakan akses dengan ID organisasi, bukan kebijakan akses yang mencantumkan ID AWS akun tertentu, sertakan `force` parameternya.

> **Warning**
>
> Jika Anda bekerja dengan log yang dikirim oleh AWS layanan yang terdaftar di[Aktifkan pencatatan dari AWS layanan](), maka sebelum melakukan langkah ini, Anda harus terlebih dahulu memperbarui filter langganan di semua akun pengirim seperti yang dijelaskan di[Langkah 1: Perbarui filter berlangganan]().

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
```

```
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Langganan tingkat akun lintas wilayah lintas akun menggunakan Firehose

Untuk berbagi data log lintas akun, Anda perlu membuat pengirim dan penerima data log:

- Pengirim data log —mendapatkan informasi tujuan dari penerima dan memberi tahu CloudWatch Log bahwa ia siap untuk mengirim peristiwa lognya ke tujuan yang ditentukan. Dalam prosedur di bagian lainnya, pengirim data log ditampilkan dengan nomor AWS akun fiksi 111111111111.
- Penerima data log —menyiapkan tujuan yang merangkum aliran Kinesis Data Streams dan CloudWatch memberi tahu Log bahwa penerima ingin menerima data log. Penerima kemudian membagikan informasi tentang tujuan ini dengan pengirim. Dalam prosedur di bagian lainnya, penerima data log ditampilkan dengan nomor AWS akun fiksi 22222222222222.

Contoh di bagian ini menggunakan aliran pengiriman Firehose dengan penyimpanan Amazon S3. Anda juga dapat mengatur aliran pengiriman Firehose dengan pengaturan berbeda. Untuk informasi selengkapnya, lihat Membuat Aliran Pengiriman Firehose.

> **Note**
>
> Grup log dan tujuan harus berada di AWS Wilayah yang sama. Namun, sumber daya AWS yang ditunjuk oleh tujuan dapat berada di Wilayah yang berbeda.

> **Note**
>
> Filter langganan Firehose untuk akun yang sama dan aliran pengiriman lintas wilayah didukung.

Topik

- Langkah 1: Buat aliran pengiriman Firehose
- Langkah 2: Buat tujuan
- Langkah 3: Buat kebijakan filter langganan tingkat akun
- Memvalidasi alur peristiwa log

- Memodifikasi keanggotaan tujuan saat runtime

Langkah 1: Buat aliran pengiriman Firehose

> **Important**
>
> Sebelum Anda menyelesaikan langkah-langkah berikut, Anda harus menggunakan kebijakan akses, sehingga Firehose dapat mengakses bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat Mengontrol Akses di Panduan Pengembang Amazon Data Firehose. Semua langkah di bagian ini (Langkah 1) harus dilakukan di akun penerima data log. US East (N. Virginia) digunakan dalam contoh perintah berikut. Ganti Wilayah ini dengan Wilayah yang benar untuk penerapan Anda.

Untuk membuat aliran pengiriman Firehose yang akan digunakan sebagai tujuan

1. Buat bucket Amazon S3:

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. Buat peran IAM yang memberikan izin Firehose untuk memasukkan data ke dalam bucket.

   a. Pertama, gunakan editor teks untuk membuat kebijakan kepercayaan dalam file `~/TrustPolicyForFirehose.json`.

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

   b. Buat IAM role dengan menentukan file kebijakan kepercayaan yang baru saja Anda buat.

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

   c. Output perintah ini akan terlihat serupa dengan yang berikut ini. Catat nama peran dan ARN peran.

```
{
```

```
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan Firehose di akun Anda.

    a. Pertama, gunakan editor teks untuk membuat kebijakan izin berikut dalam file bernama`~/PermissionsForFirehose.json`. Bergantung pada kasus penggunaan Anda, Anda mungkin perlu menambahkan lebih banyak izin ke file ini.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
    }]
}
```

b. Masukkan perintah berikut untuk mengaitkan kebijakan izin yang baru saja Anda buat dengan peran IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Masukkan perintah berikut untuk membuat aliran pengiriman Firehose. Ganti *my-role-arn*dan *my-bucket-arn*dengan nilai yang benar untuk penerapan Anda.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

Outputnya akan serupa dengan yang berikut ini:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Langkah 2: Buat tujuan

> **Important**
>
> Semua langkah dalam prosedur ini harus dilakukan di akun penerima data log.

Saat tujuan dibuat, CloudWatch Log mengirimkan pesan pengujian ke tujuan atas nama akun penerima. Saat filter langganan aktif nanti, CloudWatch Log mengirimkan peristiwa log ke tujuan atas nama akun sumber.

Untuk membuat tujuan

1. Tunggu hingga aliran Firehose yang Anda buat Langkah 1: Buat aliran pengiriman Firehose menjadi aktif. Anda dapat menggunakan perintah berikut untuk memeriksa StreamDescription. StreamStatusproperti.

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
```

Selain itu, perhatikan DeliveryStreamDescription. DeliveryStreamNilai ARN, karena Anda harus menggunakannya di langkah selanjutnya. Contoh output dari perintah ini:

```
{
    "DeliveryStreamDescription": {
        "DeliveryStreamName": "my-delivery-stream",
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:222222222222:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamEncryptionConfiguration": {
            "Status": "DISABLED"
        },
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
```

```
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Mungkin diperlukan satu atau dua menit bagi aliran pengiriman Anda untuk muncul dalam keadaan aktif.

2. Saat aliran pengiriman aktif, buat peran IAM yang akan memberikan izin kepada CloudWatch Log untuk memasukkan data ke aliran Firehose Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file ~/ TrustPolicyFor CWL.json. Gunakan editor teks untuk membuat kebijakan ini. Untuk informasi selengkapnya tentang titik akhir CloudWatch Log, lihat titik akhir dan CloudWatch kuota Amazon Logs.

   Kebijakan ini mencakup kunci konteks kondisi `aws:SourceArn` global yang menentukan `sourceAccountId` untuk membantu mencegah masalah keamanan wakil yang membingungkan. Jika Anda belum mengetahui ID akun sumber pada panggilan pertama, kami sarankan Anda memasukkan ARN tujuan di bidang ARN sumber. Dalam panggilan berikutnya, Anda harus mengatur ARN sumber menjadi ARN sumber sebenarnya yang Anda kumpulkan dari panggilan pertama. Untuk informasi selengkapnya, lihat Pencegahan Deputi Bingung.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
```

```
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
    }
}
```

3. Gunakan perintah aws iam create-role untuk membuat IAM role, dengan menentukan file kebijakan kepercayaan yang baru saja Anda buat.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

Berikut ini adalah contoh output. Perhatikan nilai `Role.Arn` yang dikembalikan, karena Anda akan perlu menggunakannya di langkah berikutnya.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2023-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
```

```
}
```

4. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan CloudWatch Log di akun Anda. Pertama, gunakan editor teks untuk membuat kebijakan izin dalam file ~/ PermissionsFor CWL.json:

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Kaitkan kebijakan izin dengan peran tersebut dengan memasukkan perintah berikut:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Setelah aliran pengiriman Firehose dalam status aktif dan Anda telah membuat peran IAM, Anda dapat membuat tujuan Log. CloudWatch

   a. Langkah ini tidak akan mengaitkan kebijakan akses dengan tujuan Anda dan hanya merupakan langkah pertama dari dua langkah yang akan menyelesaikan pembuatan tujuan. Catat ARN tujuan baru yang dikembalikan di payload, karena Anda akan menggunakan ini sebagai langkah `destination.arn` selanjutnya.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
```

```
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

b. Setelah langkah sebelumnya selesai, dalam akun penerima data log (222222222222), kaitkan kebijakan akses dengan tujuan. Kebijakan ini memungkinkan akun pengirim data log (111111111111) untuk mengakses tujuan hanya di akun penerima data log (222222222222). Anda dapat menggunakan editor teks untuk memasukkan kebijakan ini ke dalam `~/AccessPolicy.json` file:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. Ini membuat kebijakan yang menentukan siapa yang memiliki akses menulis ke tujuan. Kebijakan ini harus menentukan `logs:PutSubscriptionFilter` dan `logs:PutAccountPolicy` tindakan untuk mengakses tujuan. Pengguna lintas akun akan menggunakan `PutAccountPolicy` tindakan `PutSubscriptionFilter` dan untuk mengirim peristiwa log ke tujuan.

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Langkah 3: Buat kebijakan filter langganan tingkat akun

Beralihlah ke akun pengiriman, yaitu 111111111111 dalam contoh ini. Anda sekarang akan membuat kebijakan filter langganan tingkat akun di akun pengiriman. Dalam contoh ini, filter menyebabkan

setiap peristiwa log yang berisi string ERROR di semua kecuali dua grup log dikirim ke tujuan yang sebelumnya Anda buat.

```
aws logs put-account-policy \
    --policy-name "CrossAccountFirehoseExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"DestinationArn":"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination", "FilterPattern":
 "{$.userIdentity.type = AssumedRole}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Grup log akun pengirim dan tujuan harus berada di AWS Wilayah yang sama. Namun, tujuan dapat menunjuk ke AWS sumber daya seperti aliran Firehose yang terletak di Wilayah yang berbeda.

Memvalidasi alur peristiwa log

Setelah Anda membuat filter langganan, CloudWatch Log meneruskan semua peristiwa log masuk yang cocok dengan pola filter dan kriteria pemilihan ke aliran pengiriman Firehose. Data mulai muncul di bucket Amazon S3 Anda berdasarkan interval buffer waktu yang disetel pada aliran pengiriman Firehose. Setelah waktu tertentu berlalu, Anda dapat memverifikasi data dengan memeriksa bucket Amazon S3. Untuk memeriksa bucket, masukkan perintah berikut:

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

Output perintah tersebut akan serupa dengan yang berikut ini:

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2023-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
```

```
            }
        }
    ]
}
```

Anda kemudian dapat mengambil objek tertentu dari bucket dengan memasukkan perintah berikut. Ganti nilai `key` dengan nilai yang Anda temukan di perintah sebelumnya.

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Data dalam objek Amazon S3 dikompresi dengan format gzip. Anda dapat memeriksa data mentah dari baris perintah menggunakan salah satu dari perintah berikut:

Linux:

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Memodifikasi keanggotaan tujuan saat runtime

Anda mungkin mengalami situasi ketika Anda harus menambahkan atau menghapus pengirim log dari tujuan yang Anda miliki. Anda dapat menggunakan PutDestinationPolicydan `PutAccountPolicy` tindakan di tujuan Anda dengan kebijakan akses baru. Dalam contoh berikut, akun 111111111111 yang ditambahkan sudah sebelumnya dihentikan dari mengirim data log lagi, dan akun 333333333333 diaktifkan.

1. Ambil kebijakan yang saat ini terkait dengan TestDestination tujuan dan catat: AccessPolicy

   ```
   aws logs describe-destinations \
       --destination-name-prefix "testFirehoseDestination"
   ```

   Data yang dikembalikan mungkin terlihat seperti ini.

   ```
   {
   ```

```
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Perbarui kebijakan agar menunjukkan bahwa akun 111111111111 dihentikan, dan akun 333333333333 diaktifkan. Letakkan kebijakan ini di file ~/ NewAccessPolicy .json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Gunakan perintah berikut untuk mengaitkan kebijakan yang ditentukan dalam NewAccessPolicyfile.json dengan tujuan:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
```

```
    --access-policy file://~/NewAccessPolicy.json
```

Ini akhirnya akan menonaktifkan log acara dari ID akun 111111111111. Log acara dari ID akun 333333333333 mulai mengalir ke tujuan segera setelah pemilik akun 333333333333 membuat filter langganan.

Pencegahan Deputi Bingung

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

Sebaiknya gunakan kunci konteks kondisi `aws:SourceArn`aws:SourceAccountaws:SourceOrgID,, dan `aws:SourceOrgPaths`global dalam kebijakan sumber daya untuk membatasi izin yang memberikan layanan lain ke sumber daya. Gunakan `aws:SourceArn` untuk mengaitkan hanya satu sumber daya dengan akses lintas layanan. Gunakan `aws:SourceAccount` untuk membiarkan sumber daya apa pun di akun itu dikaitkan dengan penggunaan lintas layanan. Gunakan `aws:SourceOrgID` untuk memungkinkan sumber daya apa pun dari akun apa pun dalam suatu organisasi dikaitkan dengan penggunaan lintas layanan. Gunakan `aws:SourceOrgPaths` untuk mengaitkan sumber daya apa pun dari akun dalam AWS Organizations jalur dengan penggunaan lintas layanan. Untuk informasi selengkapnya tentang menggunakan dan memahami jalur, lihat Memahami jalur AWS Organizations entitas.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan keduanya `aws:SourceAccount` dan `aws:SourceArn` untuk membatasi izin.

Untuk melindungi dari masalah wakil yang membingungkan dalam skala besar, gunakan kunci konteks kondisi `aws:SourceOrgID` atau `aws:SourceOrgPaths` global dengan ID organisasi atau jalur organisasi sumber daya dalam kebijakan berbasis sumber daya Anda. Kebijakan yang menyertakan `aws:SourceOrgID` atau `aws:SourceOrgPaths` kunci akan secara otomatis menyertakan akun yang benar dan Anda tidak perlu memperbarui kebijakan secara manual saat menambahkan, menghapus, atau memindahkan akun di organisasi Anda.

Kebijakan yang didokumentasikan untuk memberikan akses ke CloudWatch Log untuk menulis data ke Kinesis Data Streams dan Firehose Langkah 1: Buat tujuan di Langkah 2: Buat tujuan dan menunjukkan bagaimana Anda dapat menggunakan awsSourceArn : global condition context key untuk membantu mencegah masalah deputi yang membingungkan.

Pencegahan rekursi log

Ada risiko menyebabkan rekursi log tak terbatas dengan filter langganan yang dapat menyebabkan peningkatan besar dalam penagihan konsumsi di CloudWatch Log dan tujuan Anda, jika tidak dicegah. Hal ini dapat terjadi ketika filter langganan dikaitkan dengan grup log yang menerima peristiwa log sebagai hasil dari alur kerja pengiriman langganan Anda. Log yang tertelan ke dalam grup log akan dikirim ke tujuan, menyebabkan grup log menelan lebih banyak log yang kemudian akan diteruskan lagi ke tujuan, membuat loop rekursi.

Misalnya, pertimbangkan filter langganan dengan tujuan sebagai Firehose, yang mengirimkan peristiwa log ke Amazon S3. Selain itu, ada juga fungsi Lambda yang memproses peristiwa baru yang dikirim ke Amazon S3 dan menghasilkan beberapa log itu sendiri. Jika filter langganan diterapkan ke grup log fungsi Lambda, maka peristiwa log yang dihasilkan oleh fungsi akan diteruskan ke Firehose dan Amazon S3 di tujuan, yang kemudian akan memanggil fungsi lagi, menyebabkan lebih banyak log diproduksi dan diteruskan ke Firehose dan Amazon S3, menyebabkan pemanggilan fungsi lainnya dan seterusnya. Ini akan terjadi dalam loop tak terbatas, yang mengarah ke peningkatan penagihan tak terduga pada konsumsi log, Firehose, dan Amazon S3.

Jika fungsi Lambda dilampirkan ke VPC dengan log aliran diaktifkan untuk Log, maka grup CloudWatch log VPC dapat menyebabkan rekursi log juga.

Kami menyarankan agar Anda tidak menerapkan filter langganan ke grup log yang merupakan bagian dari alur kerja pengiriman langganan Anda. Untuk filter langganan tingkat akun, gunakan `selectionCriteria` parameter di `PutAccountPolicy` API untuk mengecualikan grup log ini dari kebijakan.

Saat mengecualikan grup log, pertimbangkan AWS layanan berikut yang menghasilkan log dan mungkin menjadi bagian dari alur kerja pengiriman langganan Anda:

- Amazon EC2 dengan Fargate
- Lambda
- AWS Step Functions
- Log aliran VPC Amazon yang diaktifkan untuk Log CloudWatch

> **Note**
>
> Peristiwa log yang dihasilkan oleh grup log tujuan Lambda tidak akan diteruskan kembali ke fungsi Lambda untuk kebijakan filter langganan tingkat akun. Dalam hal ini, tidak termasuk grup log fungsi Lambda tujuan yang `selectionCriteria` digunakan tidak diperlukan untuk kebijakan berlangganan akun.

Filter sintaks pola untuk filter metrik, filter langganan, peristiwa log filter, dan Live Tail

> **Note**
>
> Untuk informasi tentang cara menanyakan grup log Anda dengan bahasa kueri Amazon CloudWatch Logs Insights, lihatCloudWatch Sintaks kueri Log Insights.

Dengan CloudWatch Log, Anda dapat menggunakan filter metrik untuk mengubah data log menjadi metrik yang dapat ditindaklanjuti, filter langganan untuk merutekan peristiwa log ke AWS layanan lain, memfilter peristiwa log untuk mencari peristiwa log, dan Live Tail untuk secara interaktif melihat log Anda secara real-time saat tertelan.

Pola filter membentuk sintaks yang digunakan oleh filter metrik, filter langganan, peristiwa log filter, dan Live Tail untuk mencocokkan istilah dalam peristiwa log. Istilah dapat berupa kata, frasa yang tepat, atau nilai numerik. Ekspresi reguler (regex) dapat digunakan untuk membuat pola filter mandiri, atau dapat digabungkan dengan JSON dan pola filter yang dibatasi ruang.

Buat pola filter dengan istilah yang ingin Anda cocokkan. Pola filter hanya mengembalikan peristiwa log yang berisi istilah yang Anda tentukan. Anda dapat menguji pola filter di CloudWatch konsol.

Topik

Sintaks ekspresi reguler (regex) yang didukung

Sintaks regex yang didukung

Saat menggunakan regex untuk mencari dan memfilter data log, Anda harus mengelilingi ekspresi Anda dengan. %

Pola filter dengan regex hanya dapat mencakup yang berikut:

- Karakter alfanumerik — Karakter alfanumerik adalah karakter yang berupa huruf (dari A ke Z atau a hingga z) atau digit (dari 0 hingga 9).
- Karakter simbol yang didukung - Ini termasuk: _ '#', '=', '@', '/', ';', ',', dan '-'. Misalnya, `%something!%` akan ditolak karena `'!'` tidak didukung.
- Operator yang didukung - Ini termasuk: ^ '$', '?', '[', ']', '{', '}', '|', '\', '*', '+', dan `'.'`.

`)`Operator `(` dan tidak didukung. Anda tidak dapat menggunakan tanda kurung untuk mendefinisikan subpola.

Karakter multi-byte tidak didukung.

> **Note**
>
> Kuota
> Ada maksimal 5 pola filter yang berisi regex untuk setiap grup log saat membuat filter metrik atau filter langganan.
> Ada batas 2 regex untuk setiap pola filter saat membuat pola filter terbatas atau JSON untuk filter metrik dan filter langganan atau saat memfilter peristiwa log atau Live Tail.

Penggunaan operator yang didukung

- ^: Jangkar pertandingan ke awal string. Misalnya, `%^[hc]at%` cocok dengan “topi” dan “kucing”, tetapi hanya di awal tali.
- $: Jangkar korek api ke ujung string. Misalnya, `%[hc]at$%` cocok dengan “topi” dan “kucing”, tetapi hanya di ujung tali.
- ?: Cocokkan nol atau lebih contoh dari istilah sebelumnya. Misalnya, `%colou?r%` dapat mencocokkan “warna” dan “warna”.
- `[]`: Mendefinisikan kelas karakter. Cocokkan daftar karakter atau rentang karakter yang terkandung dalam tanda kurung. Misalnya, `%[abc]%` cocok dengan “a”, “b”, atau “c”; `%[a-z]%` cocok dengan huruf kecil dari “a” ke “z”; dan `%[abcx-z]%` cocok dengan “a”, “b”, “c”, “x”, “y”, atau “z”.
- `{m, n}`: Cocokkan istilah sebelumnya setidaknya m dan tidak lebih dari n kali. Misalnya, hanya `%a{3,5}%` cocok dengan “aaa”, “aaaa”, dan “aaaaa”.

> **Note**
>
> Entah m atau n dapat dihilangkan jika Anda memilih untuk tidak menentukan minimum atau maksimum.

- `|`: Boolean “Atau”, yang cocok dengan istilah di kedua sisi bilah vertikal. Misalnya, `%gra|ey%` bisa cocok dengan “abu-abu” atau “abu-abu”.

> **Note**
>
> Sebuah istilah adalah sebagai karakter tunggal atau kelas karakter berulang yang menggunakan salah satu operator berikut:`?`,, `*`+, atau`{n,m}`.

- `\`: Karakter melarikan diri, yang memungkinkan Anda untuk menggunakan arti literal dari operator alih-alih makna khusus. Misalnya, `%\[.\]%` cocok dengan karakter tunggal yang dikelilingi oleh “[” dan “]” karena tanda kurung diloloskan, seperti “[a]”, “[b]”, “[7]”, “[@]”, “[]]”, dan “[]”.

> **Note**
>
> `%10\.10\.0\.1%`adalah cara yang benar untuk membuat regex agar sesuai dengan alamat IP 10.10.0.1.

- `*`: Cocokkan nol atau lebih contoh dari istilah sebelumnya. Misalnya, `%ab*c%` dapat mencocokkan “ac”, “abc”, dan “abbbc”; `%ab[0-9]*%` dapat mencocokkan “ab”, “ab0", dan “ab129".
- `+`: Cocokkan satu atau lebih contoh dari istilah sebelumnya. Misalnya, `%ab+c%` dapat mencocokkan “abc”, “abbc”, dan “abbbc”, tetapi tidak “ac”.
- `.`: Cocokkan karakter tunggal apa pun. Misalnya, `%.at%` mencocokkan tiga string karakter yang diakhiri dengan “at”, termasuk “hat”, “cat”, “bat”, “4at”, "#at" dan "at” (dimulai dengan spasi).

> **Note**
>
> Saat membuat regex agar sesuai dengan alamat IP, penting untuk melarikan diri dari operator. `.` Misalnya, `%10.10.0.1%` dapat mencocokkan “10010,051" yang mungkin bukan tujuan sebenarnya dari ekspresi tersebut.

- `\d`,`\D`: Cocokkan karakter digit/non-digit. Misalnya, `%\d%` setara dengan `%[0-9]%` dan `%\D%` setara dengan`%[^0-9]%`.

> **Note**
>
> Operator huruf besar menunjukkan kebalikan dari rekan huruf kecil.

- `\s`,`\S`: Cocokkan karakter spasi/karakter non-spasi putih.

> **Note**
>
> Operator huruf besar menunjukkan kebalikan dari rekan huruf kecil. Karakter spasi termasuk karakter tab (`\t`), spasi ( ), dan baris baru (`\n`).

- `\w`,`\W`: Cocokkan karakter alfanumerik/karakter non-alfanumerik. Misalnya, `%\w%` setara dengan `%[a-zA-Z_0-9]%` dan `%\W%` setara dengan`%[^a-zA-Z_0-9]%`.

> **Note**
>
> Operator huruf besar menunjukkan kebalikan dari rekan huruf kecil.

- `\xhh`: Cocokkan pemetaan ASCII untuk karakter heksadesimal dua digit. `\x`adalah urutan escape yang menunjukkan bahwa karakter berikut mewakili nilai heksadesimal untuk ASCII. hhmenentukan dua digit heksadesimal (0-9 dan A-F) yang menunjuk ke karakter dalam tabel ASCII.

> **Note**
>
> Anda dapat menggunakan `\xhh` untuk mencocokkan karakter simbol yang tidak didukung oleh pola filter. Misalnya, `%\x3A%` pertandingan`:`; dan `%\x28%` pertandingan`(`.

Menggunakan pola filter untuk mencocokkan istilah dengan ekspresi reguler (regex)

Ketentuan kecocokan menggunakan regex

Anda dapat mencocokkan istilah dalam peristiwa log Anda menggunakan pola regex yang dikelilingi dengan `%` (tanda persentase sebelum dan sesudah pola regex). Cuplikan kode berikut

menunjukkan contoh pola filter yang mengembalikan semua peristiwa log yang terdiri dari kata kunci AUTHORIZED.

Untuk daftar ekspresi reguler yang didukung, lihat Ekspresi reguler yang didukung.

```
%AUTHORIZED%
```

Pola filter ini mengembalikan pesan peristiwa log, seperti berikut ini:

- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[SUCCESS 200] AUTHORIZED REQUEST`

Menggunakan pola filter untuk mencocokkan istilah dalam peristiwa log tidak terstruktur

Ketentuan kecocokan dalam peristiwa log tidak terstruktur

Contoh berikut berisi cuplikan kode yang menunjukkan bagaimana Anda dapat menggunakan pola filter untuk mencocokkan istilah dalam peristiwa log tidak terstruktur.

> **Note**
>
> Pola filter peka huruf besar/kecil. Lampirkan frasa dan istilah yang tepat yang menyertakan karakter non-alfanumerik dalam tanda kutip ganda (“”).

Example: Match a single term

Cuplikan kode berikut menunjukkan contoh pola filter jangka tunggal yang mengembalikan semua peristiwa log di mana pesan berisi kata ERROR.

```
ERROR
```

Pola filter ini cocok dengan pesan peristiwa log, seperti berikut ini:

- [ERROR 400] BAD REQUEST
- [ERROR 401] UNAUTHORIZED REQUEST
- [ERROR 419] MISSING ARGUMENTS
- [ERROR 420] INVALID ARGUMENTS

Example: Match multiple terms

Cuplikan kode berikut menunjukkan contoh pola filter multi-istilah yang mengembalikan semua peristiwa log di mana pesan berisi kata-kata ERROR dan ARGUMENTS.

```
ERROR ARGUMENTS
```

Filter mengembalikan pesan peristiwa log, seperti berikut ini:

- [ERROR 419] MISSING ARGUMENTS
- [ERROR 420] INVALID ARGUMENTS

Pola filter ini tidak mengembalikan pesan peristiwa log berikut karena tidak berisi kedua istilah yang ditentukan dalam pola filter.

- [ERROR 400] BAD REQUEST
- [ERROR 401] UNAUTHORIZED REQUEST

Example: Match optional terms

Anda dapat menggunakan pencocokan pola untuk membuat pola filter yang menampilkan peristiwa log yang berisi istilah opsional. Tempatkan tanda tanya (“?”) sebelum persyaratan yang ingin Anda cocokkan. Cuplikan kode berikut menunjukkan contoh pola filter yang mengembalikan semua peristiwa log di mana pesan berisi kata ERROR atau kata ARGUMENTS.

```
?ERROR ?ARGUMENTS
```

Pola filter ini cocok dengan pesan peristiwa log, seperti berikut ini:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

> **Note**
>
> Anda tidak dapat menggabungkan tanda tanya (“?”) dengan pola filter lainnya, seperti menyertakan dan mengecualikan istilah. Jika Anda menggabungkan “?” dengan pola filter lainnya, tanda tanya (“?”) akan diabaikan.
> Misalnya, pola filter berikut cocok dengan semua peristiwa yang mengandung kata`REQUEST`, tetapi tanda tanya (“?”) filter diabaikan dan tidak berpengaruh.
>
> ```
> ?ERROR ?ARGUMENTS REQUEST
> ```
>
> Log pertandingan acara
>
> - `[INFO] REQUEST FAILED`
> - `[WARN] UNAUTHORIZED REQUEST`
> - `[ERROR] 400 BAD REQUEST`

Example: Match exact phrases

Cuplikan kode berikut menunjukkan contoh pola filter yang mengembalikan peristiwa log di mana pesan berisi frase yang tepat INTERNAL SERVER ERROR.

```
"INTERNAL SERVER ERROR"
```

Pola filter ini mengembalikan pesan peristiwa log berikut:

- `[ERROR 500] INTERNAL SERVER ERROR`

Example: Include and exclude terms

Anda dapat membuat pola filter yang menampilkan peristiwa log di mana pesan menyertakan beberapa istilah dan mengecualikan istilah lain. Tempatkan simbol minus (“-”) sebelum istilah yang ingin Anda kecualikan. Cuplikan kode berikut menunjukkan contoh pola filter yang mengembalikan peristiwa log di mana pesan menyertakan istilah ERROR dan mengecualikan istilah ARGUMEN.

```
ERROR -ARGUMENTS
```

Pola filter ini mengembalikan pesan peristiwa log, seperti berikut ini:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Pola filter ini tidak mengembalikan pesan peristiwa log berikut karena mengandung kata ARGUMENTS.

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match everything

Anda dapat mencocokkan semua yang ada di acara log Anda dengan tanda kutip ganda. Cuplikan kode berikut menunjukkan contoh pola filter yang mengembalikan semua peristiwa log.

```
" "
```

Menggunakan pola filter untuk mencocokkan istilah dalam peristiwa log JSON

Menulis pola filter untuk peristiwa log JSON

Berikut ini menjelaskan cara menulis sintaks untuk pola filter yang cocok dengan istilah JSON yang berisi string dan nilai numerik.

Writing filter patterns that match strings

Anda dapat membuat pola filter untuk mencocokkan string dalam peristiwa log JSON. Cuplikan kode berikut menunjukkan contoh sintaks untuk pola filter berbasis string.

```
{ PropertySelector EqualityOperator String }
```

Lampirkan pola filter dalam kurung kurawal (“{}”). Pola filter berbasis string harus berisi bagian-bagian berikut:

- Pemilih properti

  Matikan pemilih properti dengan tanda dolar diikuti dengan titik (“$.”). Penyeleksi properti adalah string alfanumerik yang mendukung karakter tanda hubung (“-”) dan garis bawah (“_”). String tidak mendukung notasi ilmiah. Penyeleksi properti menunjuk ke node nilai dalam peristiwa log JSON. Node nilai dapat berupa string atau angka. Tempatkan array setelah pemilih properti. Unsur-unsur dalam array mengikuti sistem penomoran berbasis nol, yang berarti bahwa elemen pertama dalam array adalah elemen 0, elemen kedua adalah elemen 1, dan seterusnya. Lampirkan elemen dalam tanda kurung (“[]”). Jika pemilih properti menunjuk ke array atau objek, pola filter tidak akan cocok dengan format log. Jika properti JSON berisi periode (`"."`), maka notasi braket dapat digunakan untuk memilih properti itu.

  > **Note**
  >
  > Pemilih wildcard
  > Anda dapat menggunakan wildcard JSON untuk memilih elemen array atau bidang objek JSON apa pun.
  > Kuota

Anda hanya dapat menggunakan hingga satu pemilih wildcard di pemilih properti.

- Operator kesetaraan

  Matikan operator kesetaraan dengan salah satu simbol berikut: sama (“=”) atau tidak sama (“!=”). Operator kesetaraan mengembalikan nilai Boolean (benar atau salah).

- Tali

  Anda dapat melampirkan string dalam tanda kutip ganda (“”). String yang berisi tipe selain karakter alfanumerik dan simbol garis bawah harus ditempatkan dalam tanda kutip ganda. Gunakan tanda bintang (“*”) sebagai kartu liar untuk mencocokkan teks.

  > **Note**
  >
  > Anda dapat menggunakan ekspresi reguler bersyarat apa pun saat membuat pola filter untuk mencocokkan istilah dalam peristiwa log JSON. Untuk daftar ekspresi reguler yang didukung, lihat Ekspresi reguler yang didukung.

Cuplikan kode berikut berisi contoh pola filter yang menunjukkan bagaimana Anda dapat memformat pola filter agar sesuai dengan istilah JSON dengan string.

```
{ $.eventType = "UpdateTrail" }
```

Writing filter patterns that match numeric values

Anda dapat membuat pola filter untuk mencocokkan nilai numerik dalam peristiwa log JSON. Cuplikan kode berikut menunjukkan contoh sintaks untuk pola filter yang cocok dengan nilai numerik.

```
{ PropertySelector NumericOperator Number }
```

Lampirkan pola filter dalam kurung kurawal (“{}”). Pola filter yang cocok dengan nilai numerik harus memiliki bagian-bagian berikut:

- Pemilih properti

  Matikan pemilih properti dengan tanda dolar diikuti dengan titik (“$.”). Penyeleksi properti adalah string alfanumerik yang mendukung karakter tanda hubung (“-”) dan garis bawah (“_”). String tidak mendukung notasi ilmiah. Penyeleksi properti menunjuk ke node nilai dalam peristiwa log JSON. Node nilai dapat berupa string atau angka. Tempatkan array setelah pemilih properti. Unsur-unsur dalam array mengikuti sistem penomoran berbasis nol, yang berarti bahwa elemen pertama dalam array adalah elemen 0, elemen kedua adalah elemen 1, dan seterusnya. Lampirkan elemen dalam tanda kurung (“[]”). Jika pemilih properti menunjuk ke array atau objek, pola filter tidak akan cocok dengan format log. Jika properti JSON berisi periode (`"."`), maka notasi braket dapat digunakan untuk memilih properti itu.

  > **Note**
  >
  > Pemilih wildcard
  > Anda dapat menggunakan wildcard JSON untuk memilih elemen array atau bidang objek JSON apa pun.
  > Kuota
  > Anda hanya dapat menggunakan hingga satu pemilih wildcard di pemilih properti.

- Operator numerik

  Matikan operator numerik dengan salah satu simbol berikut: lebih besar dari (“>”), kurang dari (“<”), sama (“=”), tidak sama (“! =”), lebih besar dari atau sama dengan (“>=”), atau kurang dari atau sama dengan (“<=”).

- Nomor

  Anda dapat menggunakan bilangan bulat yang berisi simbol plus (“+”) atau minus (“-”) dan mengikuti notasi ilmiah. Gunakan tanda bintang (“*”) sebagai kartu liar untuk mencocokkan angka.

Cuplikan kode berikut berisi contoh yang menunjukkan bagaimana Anda dapat memformat pola filter agar sesuai dengan istilah JSON dengan nilai numerik.

```
// Filter pattern with greater than symbol
{ $.bandwidth > 75 }
// Filter pattern with less than symbol
{ $.latency < 50 }
```

```
// Filter pattern with greater than or equal to symbol
{ $.refreshRate >= 60 }
// Filter pattern with less than or equal to symbol
{ $.responseTime <= 5 }
// Filter pattern with equal sign
{ $.errorCode = 400}
// Filter pattern with not equal sign
{ $.errorCode != 500 }
// Filter pattern with scientific notation and plus symbol
{ $.number[0] = 1e-3 }
// Filter pattern with scientific notation and minus symbol
{ $.number[0] != 1e+3 }
```

Ketentuan kecocokan dalam peristiwa log JSON menggunakan ekspresi sederhana

Contoh berikut berisi cuplikan kode yang menunjukkan bagaimana pola filter dapat mencocokkan istilah dalam peristiwa log JSON.

> **Note**
>
> Jika Anda menguji pola filter contoh dengan contoh peristiwa log JSON, Anda harus memasukkan log JSON contoh pada satu baris.

Peristiwa log JSON

```
{
      "eventType": "UpdateTrail",
      "sourceIPAddress": "111.111.111.111",
      "arrayKey": [
            "value",
            "another value"
      ],
      "objectList": [
           {
             "name": "a",
             "id": 1
           },
           {
             "name": "b",
             "id": 2
```

```
            }
        ],
        "SomeObject": null,
        "cluster.name": "c"
}
```

Example: Filter pattern that matches string values

Pola filter ini cocok dengan string "UpdateTrail" di properti"eventType".

```
{ $.eventType = "UpdateTrail" }
```

Example: Filter pattern that matches string values (IP address)

Pola filter ini berisi kartu liar dan cocok dengan properti "sourceIPAddress" karena tidak mengandung angka dengan awalan"123.123.".

```
{ $.sourceIPAddress != 123.123.* }
```

Example: Filter pattern that matches a specific array element with a string value

Pola filter ini cocok dengan elemen "value" dalam array"arrayKey".

```
{ $.arrayKey[0] = "value" }
```

Example: Filter pattern that matches a string using regex

Pola filter ini cocok dengan string "Trail" di properti"eventType".

```
{ $.eventType = %Trail% }
```

Example: Filter pattern that uses a wildcard to match values of any element in the array using regex

Pola filter berisi regex yang cocok dengan elemen "value" dalam array. "arrayKey"

```
{ $.arrayKey[*] = %val.{2}% }
```

Example: Filter pattern that uses a wildcard to match values of any element with a specific prefix and subnet using regex (IP address)

Pola filter ini berisi regex yang cocok dengan elemen `"111.111.111.111"` dalam properti. `"sourceIPAddress"`

```
{ $.* = %111\.111\.111\.1[0-9]{1,2}% }
```

> **Note**
>
> Kuota
> Anda hanya dapat menggunakan hingga satu pemilih wildcard di pemilih properti.

Example: Filter pattern that matches a JSON property with a period (.) in the key

```
{ $.['cluster.name'] = "c" }
```

Example: Filter pattern that matches JSON logs using ADALAH

Anda dapat membuat pola filter yang cocok dengan bidang di log JSON dengan `IS` variabel. `IS`Variabel dapat mencocokkan bidang yang berisi nilai`NULL`,`TRUE`, atau`FALSE`. Pola filter berikut mengembalikan log JSON di mana nilai `SomeObject` adalah`NULL`.

```
{ $.SomeObject IS NULL }
```

Example: Filter pattern that matches JSON logs using TIDAK ADA

Anda dapat membuat pola filter dengan `NOT EXISTS` variabel untuk mengembalikan log JSON yang tidak berisi bidang tertentu dalam data log. Pola filter berikut digunakan `NOT EXISTS` untuk mengembalikan log JSON yang tidak berisi bidang`SomeOtherObject`.

```
{ $.SomeOtherObject NOT EXISTS }
```

> **Note**
>
> Variabel `IS NOT` dan `EXISTS` saat ini tidak didukung.

Cocokkan istilah dalam objek JSON menggunakan ekspresi majemuk

Anda dapat menggunakan operator logika AND (“&&”) dan OR (“||”) dalam pola filter untuk membuat ekspresi gabungan yang cocok dengan peristiwa log di mana dua kondisi atau lebih benar. Ekspresi majemuk mendukung penggunaan tanda kurung (“()”) dan urutan operasi standar berikut: () > && > ||. Contoh berikut berisi cuplikan kode yang menunjukkan bagaimana Anda dapat menggunakan pola filter dengan ekspresi majemuk untuk mencocokkan istilah dalam objek JSON.

Objek JSON

```
{
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
         "email": "Jane.Doe@example.com"
        }
    ],
```

```
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Example: Expression that matches using AND (&&)

Pola filter ini berisi ekspresi majemuk yang cocok `"id"` `"user"` dengan nilai numerik 1 dan `"email"` dalam elemen pertama dari `"users"` array dengan string`"John.Doe@example.com"`.

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

Example: Expression that matches using OR (||)

Pola filter ini berisi ekspresi majemuk `"email"` yang cocok `"user"` dengan string`"John.Stiles@example.com"`.

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch" &&
 $.actions[2] = "nonmatch" }
```

Example: Expression that doesn't match using AND (&&)

Pola filter ini berisi ekspresi majemuk yang tidak menemukan kecocokan karena ekspresi tidak cocok dengan tindakan ketiga di`"actions"`.

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch") &&
 $.actions[2] = "nonmatch" }
```

> **Note**
>
> Kuota
> Anda hanya dapat menggunakan hingga satu pemilih wildcard di pemilih properti, dan hingga tiga pemilih wildcard dalam pola filter dengan ekspresi majemuk.

Example: Expression that doesn't match using OR (||)

Pola filter ini berisi ekspresi majemuk yang tidak menemukan kecocokan karena ekspresi tidak cocok dengan properti pertama `"users"` atau tindakan ketiga di`"actions"`.

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

Menggunakan pola filter untuk mencocokkan istilah dalam peristiwa log yang dibatasi ruang

Menulis pola filter untuk peristiwa log yang dibatasi ruang

Anda dapat membuat pola filter agar sesuai dengan istilah dalam peristiwa log yang dibatasi ruang. Berikut ini memberikan contoh peristiwa log yang dibatasi ruang dan menjelaskan cara menulis sintaks untuk pola filter yang cocok dengan istilah dalam peristiwa log yang dibatasi ruang.

> **Note**
>
> Anda dapat menggunakan ekspresi reguler bersyarat apa pun saat membuat pola filter untuk mencocokkan istilah dalam peristiwa log yang dibatasi spasi. Untuk daftar ekspresi reguler yang didukung, lihat Ekspresi reguler yang didukung.

Example: Space-delimited log event

Cuplikan kode berikut menunjukkan peristiwa log yang dibatasi spasi yang berisi tujuh bidang:`ip`,,,,,`user`, `username` dan. `timestamp` `request` `status_code` `bytes`

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

> **Note**
>
> Karakter antara tanda kurung (“[]”) dan tanda kutip ganda (“”) dianggap bidang tunggal.

Writing filter patterns that match terms in a space-delimited log event

Untuk membuat pola filter yang cocok dengan istilah dalam peristiwa log yang dibatasi spasi, lampirkan pola filter dalam tanda kurung (“[]”), dan tentukan bidang dengan nama yang dipisahkan dengan koma (“,”). Pola filter berikut mem-parsing tujuh bidang.

```
[ip=%127\.0\.0\.[1-9]%, user, username, timestamp, request =*.html*, status_code =
 4*, bytes]
```

Anda dapat menggunakan operator numerik (**>**, **<**, **=**,**!** **=**, **>=**, atau **<** **=**) dan tanda bintang (*) sebagai wild card atau regex untuk memberikan kondisi pola filter Anda. Dalam contoh pola filter, `ip` menggunakan regex yang cocok dengan rentang alamat IP 127.0.0.1 - 127.0.0.9, `request` berisi wildcard yang menyatakan harus mengekstrak nilai dengan`.html`, dan `status_code` berisi wildcard yang menyatakan harus mengekstrak nilai yang dimulai dengan. `4`

Jika Anda tidak mengetahui jumlah bidang yang Anda parsing dalam peristiwa log yang dibatasi spasi, Anda dapat menggunakan ellipsis (...) untuk mereferensikan bidang yang tidak disebutkan namanya. Elipsis dapat mereferensikan bidang sebanyak yang diperlukan. Contoh

berikut menunjukkan pola filter dengan elipsis yang mewakili empat bidang pertama yang tidak disebutkan namanya yang ditunjukkan pada pola filter contoh sebelumnya.

```
[..., request =*.html*, status_code = 4*, bytes]
```

Anda juga dapat menggunakan operator logika AND (&&) dan OR (||) untuk membuat ekspresi majemuk. Pola filter berikut berisi ekspresi majemuk yang menyatakan nilai `status_code` must be `404` atau`410`.

```
[ip, user, username, timestamp, request =*.html*, status_code = 404 || status_code =
 410, bytes]
```

Ketentuan kecocokan dalam peristiwa log yang dibatasi ruang menggunakan pencocokan pola

Anda dapat menggunakan pencocokan pola untuk membuat pola filter yang dibatasi ruang yang cocok dengan istilah dalam urutan tertentu. Tentukan urutan persyaratan Anda dengan indikator. Gunakan w1 untuk mewakili istilah pertama Anda dan w2 dan seterusnya untuk mewakili urutan persyaratan Anda berikutnya. Tempatkan koma (“,”) di antara istilah Anda. Contoh berikut berisi cuplikan kode yang menunjukkan bagaimana Anda dapat menggunakan pencocokan pola dengan pola filter yang dibatasi spasi.

> **Note**
>
> Anda dapat menggunakan ekspresi reguler bersyarat apa pun saat membuat pola filter untuk mencocokkan istilah dalam peristiwa log yang dibatasi spasi. Untuk daftar ekspresi reguler yang didukung, lihat Ekspresi reguler yang didukung.

Peristiwa log yang dibatasi ruang

```
INFO 09/25/2014 12:00:00 GET /service/resource/67 1200
INFO 09/25/2014 12:00:01 POST /service/resource/67/part/111 1310
WARNING 09/25/2014 12:00:02 Invalid user request
```

```
ERROR 09/25/2014 12:00:02 Failed to process request
```

Example: Match terms in order

Pola filter yang dibatasi spasi berikut mengembalikan peristiwa log di mana kata pertama dalam peristiwa log adalah ERROR.

```
[w1=ERROR, w2]
```

> **Note**
>
> Saat Anda membuat pola filter yang dibatasi spasi yang menggunakan pencocokan pola, Anda harus menyertakan indikator kosong setelah Anda menentukan urutan istilah Anda. Misalnya, jika Anda membuat pola filter yang mengembalikan peristiwa log di mana kata pertama adalah ERROR, sertakan indikator w2 kosong setelah istilah w1.

Example: Match terms with AND (&&) and OR (||)

Anda dapat menggunakan operator logis AND ("&&") dan OR ("||") untuk membuat pola filter yang dibatasi spasi yang berisi kondisi. Pola filter berikut mengembalikan peristiwa log di mana kata pertama dalam peristiwa adalah ERROR atau PERINGATAN.

```
[w1=ERROR || w1=WARNING, w2]
```

Example: Exclude terms from matches

Anda dapat membuat pola filter yang dibatasi spasi yang menampilkan peristiwa log tidak termasuk satu atau beberapa istilah. Tempatkan simbol yang tidak sama ("! =") sebelum istilah atau istilah yang ingin Anda kecualikan. Cuplikan kode berikut menunjukkan contoh pola filter yang mengembalikan peristiwa log di mana kata-kata pertama tidak ERROR dan PERINGATAN.

```
[w1!=ERROR && w1!=WARNING, w2]
```

Example: Match the top level item in a resource URI

Cuplikan kode berikut menunjukkan contoh pola filter yang cocok dengan item tingkat atas dalam URI sumber daya menggunakan regex.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+$%, response_time]
```

Example: Match the child level item in a resource URI

Cuplikan kode berikut menunjukkan contoh pola filter yang cocok dengan item tingkat anak dalam URI sumber daya menggunakan regex.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+/part/[0-9]+$%,
 response_time]
```

Aktifkan pencatatan dari AWS layanan

Meskipun banyak layanan mempublikasikan log hanya ke CloudWatch Log, beberapa AWS layanan dapat mempublikasikan log langsung ke Amazon Simple Storage Service atau Amazon Data Firehose. Jika persyaratan utama Anda untuk log adalah penyimpanan atau pemrosesan di salah satu layanan ini, Anda dapat dengan mudah memiliki layanan yang menghasilkan log mengirimkannya langsung ke Amazon S3 atau Firehose tanpa pengaturan tambahan.

Bahkan ketika log dipublikasikan langsung ke Amazon S3 atau Firehose, biaya berlaku. Untuk informasi selengkapnya, lihat Log Terjual di tab Log di CloudWatch Harga Amazon.

Beberapa AWS layanan menggunakan infrastruktur umum untuk mengirim log mereka. Untuk mengaktifkan logging dari layanan ini, Anda harus masuk sebagai pengguna yang memiliki izin tertentu. Selain itu, Anda harus memberikan izin AWS untuk mengaktifkan log yang akan dikirim.

Untuk layanan yang memerlukan izin ini, ada dua versi izin yang diperlukan. Layanan yang memerlukan izin tambahan ini dicatat sebagai [Izin V1] yang Didukung dan [Izin V2] yang Didukung dalam tabel. Untuk informasi tentang izin yang diperlukan ini, lihat bagian setelah tabel.

Jenis log	CloudWatch Logs	Amazon S3	Firehose
Log akses Amazon API Gateway	Didukung [Izin V1]		
AWS AppSync log	Didukung		
Log MySQL Amazon Aurora	Didukung		
Amazon Bedrock Penebangan basis pengetahuan	Didukung [Izin V2]	Didukung [Izin V2]	Didukung [Izin V2]
Log metrik kualitas media Amazon Chime dan log pesan SIP	Didukung [Izin V1]		
CloudFront: log akses		Didukung [Izin V1]	
AWS CloudHSM log audit	Didukung		

Jenis log	CloudWatch Logs	Amazon S3	Firehose
CloudWatch Terbukti evaluasi log peristiwa	Didukung [Izin V1]	Didukung [Izin V1]	
CloudWatch Log Monitor Internet		Didukung [Izin V1]	
CloudTrail log	Didukung		
AWS CodeBuild log	Didukung		
Amazon CodeWhisperer log peristiwa	Didukung [Izin V2]	Didukung [Izin V2]	Didukung [Izin V2]
Amazon Cognito log	Didukung [Izin V1]		
Log Amazon Connect	Didukung		
AWS DataSync log	Didukung		
Amazon ElastiCache untuk log Redis	Didukung [Izin V1]		Didukung [Izin V1]
AWS Elastic Beanstalk log	Didukung		
Log Layanan Kontainer Elastis Amazon	Didukung		
Log bidang kontrol Amazon Elastic Kubernetes Service	Didukung		
Amazon EventBridge Penebangan pipa	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
AWS Fargate log	Didukung		
AWS Fault Injection Service log percobaan		Didukung [Izin V1]	

Jenis log	CloudWatch Logs	Amazon S3	Firehose
Amazon FinSpace	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
AWS Global Accelerator log aliran		Didukung [Izin V1]	
AWS Glue log pekerjaan	Didukung		
Log kesalahan Pusat Identitas IAM	Didukung [Izin V2]	Didukung [Izin V2]	Didukung [Izin V2]
Log obrolan Layanan Video Interaktif Amazon	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
AWS IoT log	Didukung		
AWS IoT FleetWise log	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
AWS Lambda log	Didukung		
Log Amazon Macie	Didukung		
AWS Mainframe Modernization	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Layanan Dikelola Amazon untuk log Prometheus	Didukung [Izin V1]		
Log broker MSK Amazon	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Log Amazon MSK Connect	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Log umum dan audit Amazon MQ	Didukung		

Jenis log	CloudWatch Logs	Amazon S3	Firehose
AWS Log Firewall Jaringan	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Log akses Network Load Balancer		Didukung [Izin V1]	
OpenSearch log	Didukung		
OpenSearch Log konsumsi Layanan Amazon	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
AWS OpsWorks log	Didukung		
Log ServicePostgre SQL Database Relasional Amazon	Didukung		
AWS RoboMaker log	Didukung		
Amazon Route 53 log kueri DNS publik	Didukung		
Log kueri penyelesai Amazon Route 53	Didukung [Izin V1]	Didukung [Izin V1]	
SageMaker Acara Amazon	Didukung [Izin V1]		
Acara SageMaker pekerja Amazon	Didukung [Izin V1]		
AWS Log VPN situs-to_site	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Log Layanan Pemberitahuan Sederhana Amazon	Didukung		
Log kebijakan perlindungan data Amazon Simple Notification Service	Didukung		

Jenis log	CloudWatch Logs	Amazon S3	Firehose
File umpan data Instans Spot EC2		Didukung [Izin V1]	
AWS Step Functions Alur Kerja Ekspres dan Log Alur Kerja Standar	Didukung [Izin V1]		
Log audit Storage Gateway dan log kesehatan	Didukung [Izin V1]		
AWS Transfer Family log	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Akses Terverifikasi AWS log	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Log aliran Amazon Virtual Private Cloud	Didukung	Didukung [Izin V1]	Didukung [Izin V1]
Log akses Amazon VPC Lattice	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
AWS WAF log	Didukung [Izin V1]	Didukung [Izin V1]	Didukung
Amazon WorkMail log	Didukung [Izin V2]	Didukung [Izin V2]	Didukung [Izin V2]

Logging yang membutuhkan izin tambahan [V1]

Beberapa AWS layanan menggunakan infrastruktur umum untuk mengirim log mereka ke CloudWatch Log, Amazon S3, atau Firehose. Untuk mengaktifkan layanan AWS yang tercantum dalam tabel berikut untuk mengirim log mereka ke tujuan ini, Anda harus masuk sebagai pengguna yang memiliki izin tertentu.

Selain itu, izin harus diberikan AWS untuk mengaktifkan log yang akan dikirim. AWS dapat secara otomatis membuat izin tersebut ketika log disiapkan, atau Anda dapat membuatnya sendiri terlebih

dahulu sebelum Anda mengatur logging. Untuk pengiriman lintas akun, Anda harus membuat sendiri kebijakan izin secara manual.

Jika Anda memilih untuk AWS secara otomatis mengatur izin dan kebijakan sumber daya yang diperlukan saat Anda atau seseorang di organisasi Anda pertama kali mengatur pengiriman log, maka pengguna yang menyiapkan pengiriman log harus memiliki izin tertentu, seperti yang dijelaskan nanti di bagian ini. Selain itu, Anda dapat membuat kebijakan sumber daya sendiri, dan kemudian pengguna yang mengatur pengiriman log tidak memerlukan banyak izin.

Tabel berikut meringkas jenis log dan tujuan log mana yang terkait dengan informasi dalam bagian ini.

Bagian berikut menyediakan detail selengkapnya untuk setiap tujuan ini.

Log dikirim ke CloudWatch Log

> **Important**
>
> Ketika Anda mengatur jenis log dalam daftar berikut untuk dikirim ke CloudWatch Log, AWS membuat atau mengubah kebijakan sumber daya yang terkait dengan grup log yang menerima log, jika diperlukan. Lanjutkan membaca bagian ini untuk melihat detailnya.

Bagian ini berlaku ketika jenis log yang tercantum dalam tabel di bagian sebelumnya dikirim ke CloudWatch Log:

Izin pengguna

Untuk dapat mengatur pengiriman salah satu jenis log ini ke CloudWatch Log untuk pertama kalinya, Anda harus masuk ke akun dengan izin berikut.

- `logs:CreateLogDelivery`
- `logs:PutResourcePolicy`
- `logs:DescribeResourcePolicies`
- `logs:DescribeLogGroups`

> **Note**
>
> Saat Anda menentukan`logs:DescribeLogGroups`,`logs:DescribeResourcePolicies`, atau

`logs:PutResourcePolicy` izin, pastikan untuk mengatur ARN `Resource` barisnya untuk menggunakan * wildcard, alih-alih hanya menentukan satu nama grup log. Misalnya, `"Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*"`

Jika salah satu jenis log ini sudah dikirim ke grup CloudWatch log di Log, maka untuk mengatur pengiriman salah satu jenis log ini ke grup log yang sama, Anda hanya perlu `logs:CreateLogDelivery` izin.

Kebijakan sumber daya grup log

Grup log tempat log dikirim harus memiliki kebijakan sumber daya yang mencakup izin tertentu. Jika grup log saat ini tidak memiliki kebijakan sumber daya, dan pengguna yang mengatur logging memiliki`logs:PutResourcePolicy`,`logs:DescribeResourcePolicies`, dan `logs:DescribeLogGroups` izin untuk grup log, maka AWS secara otomatis membuat kebijakan berikut untuk itu ketika Anda mulai mengirim CloudWatch log ke Log.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "delivery.logs.amazonaws.com"
        ]
      },
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:0123456789:log-group:my-log-group:log-stream:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
```

```
      }
    }
  ]
}
```

Jika grup log memiliki kebijakan sumber daya tetapi kebijakan tersebut tidak berisi pernyataan yang ditampilkan dalam kebijakan sebelumnya, dan pengguna yang mengatur pencatatan memiliki izin `logs:PutResourcePolicy`, `logs:DescribeResourcePolicies`, dan `logs:DescribeLogGroups` untuk grup log, pernyataan tersebut ditambahkan ke kebijakan sumber daya grup log.

Pertimbangan batas ukuran kebijakan sumber daya grup log

Layanan ini harus mencantumkan setiap grup log tempat mereka mengirim log dalam kebijakan sumber daya, dan kebijakan sumber daya CloudWatch Log dibatasi hingga 5120 karakter. Layanan yang mengirimkan log ke sejumlah besar grup log mungkin mengalami batas ini.

Untuk mengurangi hal ini, CloudWatch Log memantau ukuran kebijakan sumber daya yang digunakan oleh layanan yang mengirim log, dan ketika mendeteksi bahwa kebijakan mendekati batas ukuran 5120 karakter, CloudWatch Log secara otomatis mengaktifkan `/aws/vendedlogs/*` kebijakan sumber daya untuk layanan tersebut. Anda kemudian dapat mulai menggunakan grup log dengan nama yang dimulai dengan `/aws/vendedlogs/` sebagai tujuan log dari layanan-layanan ini.

Log yang dikirim ke Amazon S3

Saat Anda menyetel log untuk dikirim ke Amazon S3, AWS buat atau ubah kebijakan sumber daya yang terkait dengan bucket S3 yang menerima log, jika diperlukan.

Log yang diterbitkan langsung ke Amazon S3 diterbitkan ke bucket lama yang Anda tentukan. Satu atau lebih berkas log dibuat setiap lima menit dalam bucket yang ditetapkan.

Ketika Anda mengirimkan log untuk pertama kalinya ke bucket Amazon S3, layanan yang mengirimkan log mencatat pemilik bucket untuk memastikan bahwa log dikirim hanya untuk bucket milik akun ini. Oleh karenanya, untuk mengubah pemilik bucket Amazon S3, Anda harus membuat ulang atau memperbarui langganan log di layanan asal.

> **Note**
>
> CloudFront menggunakan model izin yang berbeda dari layanan lain yang mengirim log vended ke S3. Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk mengonfigurasi pencatatan log standar dan untuk mengakses berkas log Anda.

Selain itu, jika Anda menggunakan bucket S3 yang sama untuk log CloudFront akses dan sumber log lain, mengaktifkan ACL di bucket untuk CloudFront juga memberikan izin ke semua sumber log lain yang menggunakan bucket ini.

Izin pengguna

Untuk dapat mengatur pengiriman salah satu jenis log ini ke Amazon S3 untuk pertama kalinya, Anda harus masuk ke akun dengan izin berikut.

- `logs:CreateLogDelivery`
- `S3:GetBucketPolicy`
- `S3:PutBucketPolicy`

Jika salah satu jenis log ini sudah dikirim ke bucket Amazon S3, untuk mengatur pengiriman dari salah satu jenis log ini ke bucket yang sama Anda hanya perlu memiliki izin `logs:CreateLogDelivery`.

Kebijakan sumber daya bucket S3

Bucket S3 tempat log dikirim harus memiliki kebijakan sumber daya yang mencakup izin tertentu. Jika bucket saat ini tidak memiliki kebijakan sumber daya dan pengguna yang menyiapkan logging memiliki izin `S3:GetBucketPolicy` dan `S3:PutBucketPolicy` izin untuk bucket, maka AWS secara otomatis membuat kebijakan berikut untuk itu saat Anda mulai mengirim log ke Amazon S3.

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
                },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-bucket",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
```

```
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        }
    ]
}
```

Dalam kebijakan sebelumnya, untuk`aws:SourceAccount`, tentukan daftar ID akun tempat log dikirimkan ke bucket ini. Untuk`aws:SourceArn`, tentukan daftar ARN dari sumber daya yang menghasilkan log, dalam formulir`arn:aws:logs:`*source-region*`:`*source-account-id*`:*`.

Jika bucket memiliki kebijakan sumber daya tetapi kebijakan tersebut tidak berisi pernyataan yang ditampilkan di kebijakan sebelumnya, dan pengguna yang menyiapkan logging memiliki `S3:PutBucketPolicy` izin `S3:GetBucketPolicy` dan untuk bucket, pernyataan tersebut akan ditambahkan ke kebijakan sumber daya bucket.

> **Note**
>
> Dalam beberapa kasus, Anda mungkin melihat `AccessDenied` kesalahan AWS CloudTrail jika `s3:ListBucket` izin belum diberikan`delivery.logs.amazonaws.com`. Untuk menghindari kesalahan ini di CloudTrail log Anda, Anda harus memberikan `s3:ListBucket` izin `delivery.logs.amazonaws.com` dan Anda harus menyertakan `Condition`

parameter yang ditampilkan dengan `s3:GetBucketAcl` izin yang ditetapkan dalam kebijakan bucket sebelumnya. Untuk membuatnya lebih sederhana, alih-alih membuat yang baru`Statement`, Anda dapat langsung memperbarui `AWSLogDeliveryAclCheck` to be `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Enkripsi sisi server bucket Amazon S3

Anda dapat melindungi data di bucket Amazon S3 dengan mengaktifkan Enkripsi sisi server dengan kunci yang dikelola Amazon S3 (SSE-S3) atau enkripsi sisi server dengan kunci yang disimpan di (SSE-KMS). AWS KMS AWS Key Management Service Untuk informasi selengkapnya, silakan lihat Melindungi data menggunakan enkripsi sisi server.

Jika Anda memilih SSE-S3, tidak diperlukan konfigurasi tambahan. Amazon S3 menangani kunci enkripsi.

Warning

Jika Anda memilih SSE-KMS, Anda harus menggunakan kunci yang dikelola pelanggan, karena menggunakan kunci AWS terkelola tidak didukung untuk skenario ini. Jika Anda mengatur enkripsi menggunakan kunci AWS terkelola, log akan dikirimkan dalam format yang tidak dapat dibaca.

Saat menggunakan AWS KMS kunci terkelola pelanggan, Anda dapat menentukan Nama Sumber Daya Amazon (ARN) kunci terkelola pelanggan saat mengaktifkan enkripsi bucket. Anda harus menambahkan hal berikut ke kebijakan kunci untuk kunci terkelola pelanggan Anda (bukan ke kebijakan bucket untuk bucket S3 Anda), sehingga akun pengiriman log dapat menulis ke bucket S3 Anda.

Jika Anda memilih SSE-KMS, Anda harus menggunakan kunci yang dikelola pelanggan, karena menggunakan kunci AWS terkelola tidak didukung untuk skenario ini. Saat menggunakan AWS KMS kunci terkelola pelanggan, Anda dapat menentukan Nama Sumber Daya Amazon (ARN) kunci terkelola pelanggan saat mengaktifkan enkripsi bucket. Anda harus menambahkan hal berikut ke kebijakan kunci untuk kunci terkelola pelanggan Anda (bukan ke kebijakan bucket untuk bucket S3 Anda), sehingga akun pengiriman log dapat menulis ke bucket S3 Anda.

```
{
```

```
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
        }
}
```

Untuk`aws:SourceAccount`, tentukan daftar ID akun tempat log dikirimkan ke bucket ini. Untuk`aws:SourceArn`, tentukan daftar ARN dari sumber daya yang menghasilkan log, dalam formulir`arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Log dikirim ke Firehose

Bagian ini berlaku ketika jenis log yang tercantum dalam tabel di bagian sebelumnya dikirim ke Firehose:

Izin pengguna

Untuk dapat mengatur pengiriman salah satu jenis log ini ke Firehose untuk pertama kalinya, Anda harus masuk ke akun dengan izin berikut.

- `logs:CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam:CreateServiceLinkedRole`

Jika salah satu dari jenis log ini sudah dikirim ke Firehose, maka untuk mengatur pengiriman salah satu dari jenis log ini ke Firehose, Anda hanya perlu memiliki izin dan izin. `logs:CreateLogDelivery` `firehose:TagDeliveryStream`

Peran IAM yang digunakan untuk izin

Karena Firehose tidak menggunakan kebijakan sumber daya, AWS menggunakan peran IAM saat menyiapkan log ini untuk dikirim ke Firehose. AWS membuat peran terkait layanan bernama. AWSServiceRoleForLogDelivery Peran terkait layanan ini mencakup izin berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Peran terkait layanan ini memberikan izin untuk semua aliran pengiriman Firehose yang memiliki tag yang disetel ke. `LogDeliveryEnabled` `true` AWS memberikan tag ini ke aliran pengiriman tujuan saat Anda mengatur logging.

Peran terkait layanan ini juga memiliki kebijakan kepercayaan yang memungkinkan layanan `delivery.logs.amazonaws.com` utama untuk mengasumsikan peran yang terhubung dengan layanan yang diperlukan. Kebijakan kepercayaan tersebut adalah sebagai berikut:

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

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Logging yang membutuhkan izin tambahan [V2]

Beberapa AWS layanan menggunakan metode baru untuk mengirim log mereka. Ini adalah metode fleksibel yang memungkinkan Anda mengatur pengiriman log dari layanan ini ke satu atau beberapa tujuan berikut: CloudWatch Log, Amazon S3, atau Firehose.

Pengiriman log kerja terdiri dari tiga elemen:

- `ADeliverySource`, yang merupakan objek logis yang mewakili sumber daya yang benar-benar mengirim log.
- `ADeliveryDestination`, yang merupakan objek logis yang mewakili tujuan pengiriman yang sebenarnya.
- `ADelivery`, yang menghubungkan sumber pengiriman ke tujuan pengiriman

Untuk mengonfigurasi pengiriman log antara AWS layanan yang didukung dan tujuan, Anda harus melakukan hal berikut:

- Buat sumber pengiriman dengan PutDeliverySource.
- Buat tujuan pengiriman dengan PutDeliveryDestination.
- Jika Anda mengirimkan log lintas akun, Anda harus menggunakan PutDeliveryDestinationPolicydi akun tujuan untuk menetapkan IAM kebijakan ke tujuan. Kebijakan ini mengotorisasi pembuatan pengiriman dari sumber pengiriman di akun A ke tujuan pengiriman di akun B. Untuk pengiriman lintas akun, Anda harus membuat sendiri kebijakan izin secara manual.
- Buat pengiriman dengan memasangkan tepat satu sumber pengiriman dan satu tujuan pengiriman, dengan menggunakan CreateDelivery.

Bagian berikut memberikan rincian izin yang perlu Anda miliki saat Anda masuk untuk mengatur pengiriman log ke setiap jenis tujuan, menggunakan proses V2. Izin ini dapat diberikan ke peran IAM yang Anda masuki.

> **Important**
>
> Anda bertanggung jawab untuk menghapus sumber daya pengiriman log setelah menghapus sumber daya penghasil log. Untuk melakukannya, ikuti langkah-langkah ini.
>
> 1. Hapus `Delivery` dengan menggunakan DeleteDeliveryoperasi.
> 2. Hapus `DeliverySource` dengan menggunakan DeleteDeliverySourceoperasi.
> 3. Jika yang `DeliveryDestination` terkait dengan `DeliverySource` yang baru saja Anda hapus hanya digunakan untuk spesifik ini`DeliverySource`, maka Anda dapat menghapusnya dengan menggunakan DeleteDeliveryDestinationsoperasi.

Daftar Isi

Log dikirim ke CloudWatch Log

Izin pengguna

Untuk mengaktifkan pengiriman CloudWatch log ke Log, Anda harus masuk dengan izin berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
```

```
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyCWL",
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:*"
            ]
        }
    ]
}
```

Kebijakan sumber daya grup log

Grup log tempat log dikirim harus memiliki kebijakan sumber daya yang mencakup izin tertentu. Jika grup log saat ini tidak memiliki kebijakan sumber daya, dan pengguna yang mengatur logging memiliki`logs:PutResourcePolicy`,`logs:DescribeResourcePolicies`, dan `logs:DescribeLogGroups` izin untuk grup log, maka AWS secara otomatis membuat kebijakan berikut untuk itu ketika Anda mulai mengirim CloudWatch log ke Log.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "delivery.logs.amazonaws.com"
        ]
      },
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:0123456789:log-group:my-log-group:log-stream:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

Pertimbangan batas ukuran kebijakan sumber daya grup log

Layanan ini harus mencantumkan setiap grup log tempat mereka mengirim log dalam kebijakan sumber daya, dan kebijakan sumber daya CloudWatch Log dibatasi hingga 5120 karakter. Layanan yang mengirimkan log ke sejumlah besar grup log dapat mencapai batasan ini.

Untuk mengurangi hal ini, CloudWatch Log memantau ukuran kebijakan sumber daya yang digunakan oleh layanan yang mengirim log, dan ketika mendeteksi bahwa kebijakan mendekati batas ukuran 5120 karakter, CloudWatch Log secara otomatis mengaktifkan `/aws/vendedlogs/*` kebijakan sumber daya untuk layanan tersebut. Anda kemudian dapat mulai menggunakan grup log dengan nama yang dimulai dengan `/aws/vendedlogs/` sebagai tujuan log dari layanan-layanan ini.

Log yang dikirim ke Amazon S3

Izin pengguna

Untuk mengaktifkan pengiriman log ke Amazon S3, Anda harus masuk dengan izin berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
```

```
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

Bucket S3 tempat log dikirim harus memiliki kebijakan sumber daya yang mencakup izin tertentu. Jika bucket saat ini tidak memiliki kebijakan sumber daya dan pengguna yang menyiapkan logging memiliki izin `S3:GetBucketPolicy` dan `S3:PutBucketPolicy` izin untuk bucket, maka AWS secara otomatis membuat kebijakan berikut untuk itu saat Anda mulai mengirim log ke Amazon S3.

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
                },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-bucket",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
```

```
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source*"]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-
source:*"]
                }
            }
        }
    ]
}
```

Dalam kebijakan sebelumnya, untuk`aws:SourceAccount`, tentukan daftar ID akun tempat log dikirimkan ke bucket ini. Untuk`aws:SourceArn`, tentukan daftar ARN dari sumber daya yang menghasilkan log, dalam formulir`arn:aws:logs:`*source-region*`:`*source-account-id*`:*`.

Jika bucket memiliki kebijakan sumber daya tetapi kebijakan tersebut tidak berisi pernyataan yang ditampilkan di kebijakan sebelumnya, dan pengguna yang menyiapkan logging memiliki `S3:PutBucketPolicy` izin `S3:GetBucketPolicy` dan untuk bucket, pernyataan tersebut akan ditambahkan ke kebijakan sumber daya bucket.

> **Note**
>
> Dalam beberapa kasus, Anda mungkin melihat `AccessDenied` kesalahan AWS CloudTrail jika `s3:ListBucket` izin belum diberikan`delivery.logs.amazonaws.com`. Untuk menghindari kesalahan ini di CloudTrail log Anda, Anda harus memberikan `s3:ListBucket` izin `delivery.logs.amazonaws.com` dan Anda harus menyertakan `Condition` parameter yang ditampilkan dengan `s3:GetBucketAcl` izin yang ditetapkan dalam

kebijakan bucket sebelumnya. Untuk membuatnya lebih sederhana, alih-alih membuat yang baru`Statement`, Anda dapat langsung memperbarui `AWSLogDeliveryAclCheck` to be `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Enkripsi sisi server bucket Amazon S3

Anda dapat melindungi data di bucket Amazon S3 dengan mengaktifkan Enkripsi sisi server dengan kunci yang dikelola Amazon S3 (SSE-S3) atau enkripsi sisi server dengan kunci yang disimpan di (SSE-KMS). AWS KMS AWS Key Management Service Untuk informasi selengkapnya, silakan lihat Melindungi data menggunakan enkripsi sisi server.

Jika Anda memilih SSE-S3, tidak diperlukan konfigurasi tambahan. Amazon S3 menangani kunci enkripsi.

Warning

Jika Anda memilih SSE-KMS, Anda harus menggunakan kunci yang dikelola pelanggan, karena menggunakan kunci AWS terkelola tidak didukung untuk skenario ini. Jika Anda mengatur enkripsi menggunakan kunci AWS terkelola, log akan dikirimkan dalam format yang tidak dapat dibaca.

Saat menggunakan AWS KMS kunci terkelola pelanggan, Anda dapat menentukan Nama Sumber Daya Amazon (ARN) kunci terkelola pelanggan saat mengaktifkan enkripsi bucket. Anda harus menambahkan hal berikut ke kebijakan kunci untuk kunci terkelola pelanggan Anda (bukan ke kebijakan bucket untuk bucket S3 Anda), sehingga akun pengiriman log dapat menulis ke bucket S3 Anda.

Jika Anda memilih SSE-KMS, Anda harus menggunakan kunci yang dikelola pelanggan, karena menggunakan kunci AWS terkelola tidak didukung untuk skenario ini. Saat menggunakan AWS KMS kunci terkelola pelanggan, Anda dapat menentukan Nama Sumber Daya Amazon (ARN) kunci terkelola pelanggan saat mengaktifkan enkripsi bucket. Anda harus menambahkan hal berikut ke kebijakan kunci untuk kunci terkelola pelanggan Anda (bukan ke kebijakan bucket untuk bucket S3 Anda), sehingga akun pengiriman log dapat menulis ke bucket S3 Anda.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
```

```
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source:*"]
        }
        }
}
```

Untuk`aws:SourceAccount`, tentukan daftar ID akun tempat log dikirimkan ke bucket ini. Untuk`aws:SourceArn`, tentukan daftar ARN dari sumber daya yang menghasilkan log, dalam formulir`arn:aws:logs:`*source-region*`:`*source-account-id*`:*`.

Log dikirim ke Firehose

Izin pengguna

Untuk mengaktifkan pengiriman log ke Firehose, Anda harus masuk dengan izin berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
```

```
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyFH",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": [
                "arn:aws:firehose:region:account-id:deliverystream/*"
            ]
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        }
    ]
```

```
}
```

Peran IAM yang digunakan untuk izin sumber daya

Karena Firehose tidak menggunakan kebijakan sumber daya, AWS menggunakan peran IAM saat menyiapkan log ini untuk dikirim ke Firehose. AWS membuat peran terkait layanan bernama. AWSServiceRoleForLogDelivery Peran terkait layanan ini mencakup izin berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Peran terkait layanan ini memberikan izin untuk semua aliran pengiriman Firehose yang memiliki tag yang disetel ke. `LogDeliveryEnabled` `true` AWS memberikan tag ini ke aliran pengiriman tujuan saat Anda mengatur logging.

Peran terkait layanan ini juga memiliki kebijakan kepercayaan yang memungkinkan layanan `delivery.logs.amazonaws.com` utama untuk mengasumsikan peran yang terhubung dengan layanan yang diperlukan. Kebijakan kepercayaan tersebut adalah sebagai berikut:

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

```
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Izin khusus layanan

Selain izin khusus tujuan yang tercantum di bagian sebelumnya, beberapa layanan memerlukan otorisasi eksplisit bahwa pelanggan diizinkan mengirim log dari sumber daya mereka, sebagai lapisan keamanan tambahan. Ini mengotorisasi `AllowVendedLogDeliveryForResource` tindakan untuk sumber daya yang menjual log dalam layanan itu. Untuk layanan ini, gunakan kebijakan berikut dan ganti *`jenis layanan dan sumber daya dengan nilai yang`* sesuai. Untuk nilai khusus layanan untuk bidang ini, lihat halaman dokumentasi layanan tersebut untuk log penjual.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceLevelAccessForLogDelivery",
            "Effect": "Allow",
            "Action": [
                "service:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:service:region:account-id:resource-type/*"
        }
    ]
}
```

Izin khusus konsol

Selain izin yang tercantum di bagian sebelumnya, jika Anda menyiapkan pengiriman log menggunakan konsol alih-alih API, Anda juga memerlukan izin tambahan berikut:

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

```
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleS3",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleFH",
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "firehose:DescribeDeliveryStream"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara

yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi aws:SourceArnaws:SourceAccountaws:SourceOrgID,, dan aws:SourceOrgPathsglobal dalam kebijakan sumber daya untuk membatasi izin yang diberikan CloudWatch Log kepada layanan lain ke sumber daya. Gunakan `aws:SourceArn` untuk mengaitkan hanya satu sumber daya dengan akses lintas layanan. Gunakan `aws:SourceAccount` untuk membiarkan sumber daya apa pun di akun itu dikaitkan dengan penggunaan lintas layanan. Gunakan `aws:SourceOrgID` untuk memungkinkan sumber daya apa pun dari akun apa pun dalam suatu organisasi dikaitkan dengan penggunaan lintas layanan. Gunakan `aws:SourceOrgPaths` untuk mengaitkan sumber daya apa pun dari akun dalam AWS Organizations jalur dengan penggunaan lintas layanan. Untuk informasi selengkapnya tentang menggunakan dan memahami jalur, lihat Memahami jalur AWS Organizations entitas.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan keduanya `aws:SourceAccount` dan `aws:SourceArn` untuk membatasi izin.

Untuk melindungi dari masalah wakil yang membingungkan dalam skala besar, gunakan kunci konteks kondisi `aws:SourceOrgID` atau `aws:SourceOrgPaths` global dengan ID organisasi atau jalur organisasi sumber daya dalam kebijakan berbasis sumber daya Anda. Kebijakan yang menyertakan `aws:SourceOrgID` atau `aws:SourceOrgPaths` kunci akan secara otomatis menyertakan akun yang benar dan Anda tidak perlu memperbarui kebijakan secara manual saat menambahkan, menghapus, atau memindahkan akun di organisasi Anda.

Kebijakan di bagian sebelumnya dari halaman ini menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global untuk mencegah masalah deputi yang membingungkan.

CloudWatch Log pembaruan ke kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk CloudWatch Log sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen CloudWatch Log.

Perubahan	Deskripsi	Tanggal
AWSServiceRoleForLogDelivery kebijakan peran terkait layanan — Pembaruan ke kebijakan yang ada	CloudWatch Log mengubah izin dalam kebijakan IAM yang terkait dengan peran terkait AWSServiceRoleForLogDeliverylayanan. Perubahan berikut dibuat: • Kunci `firehose:ResourceTag/LogDeliveryEnabled": "true"` kondisi diubah menjadi`aws:ResourceTag/LogDeliveryEnabled": "true"` .	15 Juli 2021
CloudWatch Log mulai melacak perubahan	CloudWatch Log mulai melacak perubahan untuk kebijakan yang AWS dikelola.	10 Juni 2021

Mengekspor data log ke Amazon S3

Ekspor data log dari grup log Anda ke bucket Amazon S3 dan gunakan data ini dalam pemrosesan dan analisis khusus, atau untuk memuat ke sistem lain. Anda dapat mengekspor ke ember di akun yang sama atau akun lain.

Anda dapat melakukan tindakan berikut:

- Ekspor data log ke bucket S3 yang dienkripsi oleh SSE-KMS di () AWS Key Management Service AWS KMS
- Ekspor data log ke bucket S3 yang mengaktifkan Kunci Objek S3 dengan periode retensi

> **Note**
>
> Ekspor ke Amazon S3 hanya didukung untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

Untuk memulai proses ekspor, Anda harus membuat bucket S3 untuk menyimpan data log yang diekspor. Anda dapat menyimpan file yang diekspor di bucket S3 dan menentukan aturan siklus hidup Amazon S3 untuk mengarsipkan atau menghapus file yang diekspor secara otomatis.

Anda dapat mengekspor ke bucket S3 yang dienkripsi dengan AES-256 atau dengan SSE-KMS. Mengekspor ke bucket yang dienkripsi dengan DSSE-KMS tidak didukung.

Anda dapat mengekspor log dari beberapa grup log atau beberapa rentang waktu ke bucket S3 yang sama. Untuk memisahkan data log untuk setiap tugas ekspor, Anda dapat menentukan prefiks yang akan digunakan sebagai prefiks kunci Amazon S3 untuk semua objek yang diekspor.

> **Note**
>
> Penyortiran berbasis waktu pada potongan data log di dalam file yang diekspor tidak dijamin. Anda dapat mengurutkan data bidang log yang diekspor dengan menggunakan utilitas Linux. Misalnya, perintah utilitas berikut mengurutkan peristiwa di semua `.gz` file dalam satu folder.
>
> ```
> find . -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
> ```

Perintah utilitas berikut mengurutkan file.gz dari beberapa subfolder.

```
find ./*/ -type f -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
```

Selain itu, Anda dapat menggunakan `stdout` perintah lain untuk menyalurkan output yang diurutkan ke file lain untuk menyimpannya.

Data log dapat memakan waktu hingga 12 jam agar tersedia untuk diekspor. Waktu tugas ekspor habis setelah 24 jam. Jika tugas ekspor Anda habis waktu, kurangi rentang waktu saat Anda membuat tugas ekspor.

Untuk analisis data log secara hampir waktu nyata, lihat Menganalisis data log dengan Wawasan CloudWatch Log atau Pemrosesan data log secara real-time dengan langganan.

Daftar Isi

Konsep

Sebelum Anda mulai, pahami konsep ekspor berikut:

nama grup log

Nama grup log yang terkait dengan tugas ekspor. Data log dalam grup log ini akan diekspor ke bucket S3 yang ditentukan.

dari (stempel waktu)

Stempel waktu yang diperlukan dan dinyatakan sebagai angka milidetik sejak 1 Jan 1970 00:00:00 UTC. Semua peristiwa log dalam grup log yang tertelan pada atau setelah waktu ini akan diekspor.

ke (stempel waktu)

Stempel waktu yang diperlukan dan dinyatakan sebagai angka milidetik sejak 1 Jan 1970 00:00:00 UTC. Semua log acara dalam grup log yang diserap sebelum waktu ini akan diekspor.

bucket tujuan

Nama bucket S3 yang terkait dengan tugas ekspor. Bucket ini digunakan untuk mengekspor data log dari grup log yang ditentukan.

prefiks tujuan

Atribut opsional yang digunakan sebagai key prefix Amazon S3 untuk semua objek yang diekspor. Ini membantu membuat organisasi mirip folder di bucket Anda.

Ekspor data log ke Amazon S3 menggunakan konsol

Dalam contoh berikut, Anda menggunakan CloudWatch konsol Amazon untuk mengekspor semua data dari grup CloudWatch log Amazon Logs yang diberi nama `my-log-group` ke bucket Amazon S3 bernama. `my-exported-logs`

Mengekspor data log ke bucket S3 yang dienkripsi oleh SSE-KMS didukung. Mengekspor ke bucket yang dienkripsi dengan DSSE-KMS tidak didukung.

Detail cara Anda mengatur ekspor tergantung pada apakah bucket Amazon S3 yang ingin Anda ekspor berada di akun yang sama dengan log Anda yang sedang diekspor, atau di akun lain.

Topik

Ekspor akun yang sama

Jika bucket Amazon S3 berada di akun yang sama dengan log yang sedang diekspor, gunakan instruksi di bagian ini.

Topik

Langkah 1: Buat bucket Amazon S3.

Kami menyarankan Anda menggunakan bucket yang dibuat khusus untuk CloudWatch Log. Namun, jika Anda ingin menggunakan bucket yang sudah ada, Anda dapat melompat ke langkah 2.

> **Note**
>
> Bucket S3 harus berada di Region yang sama dengan data log yang akan diekspor. CloudWatch Log tidak mendukung ekspor data ke bucket S3 di Wilayah lain.

Untuk membuat bucket S3

1. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Jika perlu, ubah Region. Dari bilah navigasi, pilih Wilayah tempat CloudWatch Log Anda berada.
3. Pilih Create Bucket (Buat Bucket).
4. Untuk Bucket Name (Nama Bucket), masukkan nama untuk bucket.
5. Untuk Wilayah, pilih Wilayah tempat data CloudWatch Log Anda berada.
6. Pilih Buat.

Langkah 2: Siapkan izin akses

Untuk membuat tugas ekspor di langkah 5, Anda harus masuk dengan peran `AmazonS3ReadOnlyAccess` IAM dan dengan izin berikut:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`

- `logs:DescribeLogGroups`

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .
- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Langkah 3: Tetapkan izin pada bucket S3

Secara default, semua bucket dan objek S3 bersifat pribadi. Hanya pemilik sumber daya, Akun AWS yang membuat ember, yang dapat mengakses ember dan objek apa pun yang dikandungnya. Namun, pemilik sumber daya dapat memilih untuk memberikan izin akses kepada sumber daya dan pengguna lain dengan menulis kebijakan akses.

Ketika Anda menetapkan kebijakan, sebaiknya Anda menyertakan string yang dihasilkan secara acak sebagai prefiks untuk bucket sehingga hanya pengaliran log yang dimaksud yang diekspor ke bucket tersebut.

> **Important**
>
> Untuk membuat ekspor ke bucket S3 lebih aman, kami sekarang meminta Anda untuk menentukan daftar akun sumber yang diizinkan untuk mengekspor data log ke bucket S3 Anda.
> Dalam contoh berikut, daftar ID akun di `aws:SourceAccount` kunci adalah akun tempat pengguna dapat mengekspor data log ke bucket S3 Anda. `aws:SourceArn`Kuncinya adalah

sumber daya tempat tindakan diambil. Anda dapat membatasi ini ke grup log tertentu, atau menggunakan wildcard seperti yang ditunjukkan dalam contoh ini.
Kami menyarankan Anda juga menyertakan ID akun akun tempat bucket S3 dibuat, untuk memungkinkan ekspor dalam akun yang sama.

Untuk mengatur izin bucket Amazon S3

1. Di konsol Amazon S3, pilih bucket yang Anda buat di langkah 1.
2. Pilih Permissions (Izin), Bucket policy (Kebijakan bucket).
3. Di Editor Kebijakan Bucket, tambahkan kebijakan berikut. Ubah `my-exported-logs` nama bucket S3 Anda. Pastikan untuk menentukan titik akhir Wilayah yang benar, seperti`us-west-1`, untuk Principal.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        }
    ]
}
```

4. Pilih Save (Simpan) untuk menetapkan kebijakan yang baru saja ditambahkan sebagai kebijakan akses di bucket Anda. Kebijakan ini memungkinkan CloudWatch Log untuk mengekspor data log ke bucket S3 Anda. Pemilik bucket memiliki izin penuh atas semua objek yang diekspor.

> **Warning**
>
> Jika bucket yang ada sudah memiliki satu atau beberapa kebijakan yang dilampirkan padanya, tambahkan pernyataan untuk akses CloudWatch Log ke kebijakan atau kebijakan tersebut. Sebaiknya Anda mengevaluasi hasil rangkaian izin untuk memastikan bahwa itu sesuai untuk pengguna yang akan mengakses bucket.

(Opsional) Langkah 4: Mengekspor ke bucket yang dienkripsi dengan SSE-KMS

Langkah ini diperlukan hanya jika Anda mengekspor ke bucket S3 yang menggunakan enkripsi sisi server. AWS KMS keys Enkripsi ini dikenal sebagai SSE-KMS.

Untuk mengekspor ke bucket yang dienkripsi dengan SSE-KMS

1. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di bilah navigasi kiri, pilih Kunci yang dikelola pelanggan.

   Pilih Buat Kunci.
4. Untuk Tipe Kunci, pilih Simetris.
5. Untuk penggunaan Kunci, pilih Enkripsi dan dekripsi dan kemudian pilih Berikutnya.
6. Di bawah Tambahkan label, masukkan alias untuk kunci dan secara opsional tambahkan deskripsi atau tag. Lalu pilih Selanjutnya.
7. Di bawah Administrator kunci, pilih siapa yang dapat mengelola kunci ini, lalu pilih Berikutnya.
8. Di bawah Tentukan izin penggunaan kunci, jangan buat perubahan dan pilih Berikutnya.
9. Tinjau pengaturan dan pilih Selesai.
10. Kembali ke halaman kunci yang dikelola Pelanggan, pilih nama kunci yang baru saja Anda buat.
11. Pilih tab Kebijakan kunci dan pilih Beralih ke tampilan kebijakan.
12. Di bagian Kebijakan kunci, pilih Edit.
13. Tambahkan pernyataan berikut ke daftar pernyataan kebijakan kunci. Ketika Anda melakukannya, ganti *Wilayah* dengan Wilayah log Anda dan ganti *akun-ARN dengan ARN* dari akun yang memiliki kunci KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

14. Pilih Simpan perubahan.
15. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
16. Temukan bucket yang Anda buat Langkah 1: Buat ember S3 dan pilih nama bucket.
17. Pilih tab Properti. Kemudian, di bawah Enkripsi Default, pilih Edit.
18. Di bawah Enkripsi sisi server, pilih Aktifkan.
19. Di bawah Tipe enkripsi memilih Kunci (SSE-KMS)AWS Key Management Service .
20. Pilih Pilih dari AWS KMS kunci Anda dan temukan kunci yang Anda buat.
21. Untuk kunci Bucket, pilih Aktifkan.
22. Pilih Simpan perubahan.

Langkah 5: Buat tugas ekspor

Di langkah ini, Anda membuat tugas ekspor untuk mengekspor log dari grup log.

Untuk mengekspor data ke Amazon S3 menggunakan konsol CloudWatch

1. Masuk dengan izin yang memadai seperti yang didokumentasikanLangkah 2: Siapkan izin akses.
2. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
3. Pada panel navigasi, pilih Grup log.
4. Di layar Log Groups (Grup Log), pilih nama grup log.

5. Pilih Actions (Tindakan), Export data to Amazon S3 (Ekspor data ke Amazon S3).
6. Di layar Export data to Amazon S3 (Ekspor data ke Amazon S3), di Define data export (Tentukan ekspor data), atur rentang waktu untuk data yang akan diekspor menggunakan From (Dari) dan To (Sampai).
7. Jika grup log Anda memiliki beberapa pengaliran log, Anda dapat memberikan prefiks pengaliran log untuk membatasi data grup log ke pengaliran tertentu. Pilih Advanced (Lanjutan), lalu untuk Stream prefix (Prefiks pengaliran), masukkan prefiks pengaliran log.
8. Di bawah bucket Pilih S3, pilih akun yang terkait dengan bucket S3.
9. Untuk nama bucket S3, pilih bucket S3.
10. Untuk S3 Bucket prefix (Prefiks bucket S3), masukkan string yang dihasilkan secara acak yang Anda tentukan dalam kebijakan bucket.
11. Pilih Export (Ekspor) untuk mengekspor data log ke Amazon S3.
12. Untuk melihat status data log yang diekspor ke Amazon S3, pilih Actions (Tindakan), lalu View all exports to Amazon S3 (Lihat semua ekspor ke Amazon S3).

Ekspor lintas akun

Jika bucket Amazon S3 berada di akun yang berbeda dari log yang sedang diekspor, gunakan petunjuk di bagian ini.

Topik

Langkah 1: Buat bucket Amazon S3.

Kami menyarankan Anda menggunakan bucket yang dibuat khusus untuk CloudWatch Log. Namun, jika Anda ingin menggunakan bucket yang sudah ada, Anda dapat melompat ke langkah 2.

> **Note**
>
> Bucket S3 harus berada di Region yang sama dengan data log yang akan diekspor. CloudWatch Log tidak mendukung ekspor data ke bucket S3 di Wilayah lain.

Untuk membuat bucket S3

1. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Jika perlu, ubah Region. Dari bilah navigasi, pilih Wilayah tempat CloudWatch Log Anda berada.
3. Pilih Create Bucket (Buat Bucket).
4. Untuk Bucket Name (Nama Bucket), masukkan nama untuk bucket.
5. Untuk Wilayah, pilih Wilayah tempat data CloudWatch Log Anda berada.
6. Pilih Buat.

Langkah 2: Siapkan izin akses

Pertama, Anda harus membuat kebijakan IAM baru untuk mengaktifkan CloudWatch Log agar memiliki `s3:PutObject` izin untuk bucket Amazon S3 tujuan di akun tujuan.

Kebijakan yang Anda buat bergantung pada apakah bucket tujuan menggunakan AWS KMS enkripsi.

Untuk membuat kebijakan IAM untuk mengekspor log ke bucket Amazon S3

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di bagian Editor kebijakan, pilih JSON.
5. Jika bucket tujuan tidak menggunakan AWS KMS enkripsi, tempelkan kebijakan berikut ke editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
```

```
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Jika bucket tujuan menggunakan AWS KMS enkripsi, tempelkan kebijakan berikut ke editor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-exported-logs/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "ARN_OF_KMS_KEY"
    }
  ]
}
```

6. Pilih Selanjutnya.
7. Masukkan nama kebijakan. Anda akan menggunakan nama ini untuk melampirkan kebijakan ke peran IAM Anda.
8. Pilih Buat kebijakan untuk menyimpan kebijakan baru.

Untuk membuat tugas ekspor di langkah 5, Anda harus masuk dengan peran `AmazonS3ReadOnlyAccess` IAM. Anda juga harus masuk dengan kebijakan IAM yang baru saja Anda buat, dan juga dengan izin berikut:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`

- `logs:DescribeLogGroups`

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .
- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Langkah 3: Tetapkan izin pada bucket S3

Secara default, semua bucket dan objek S3 bersifat pribadi. Hanya pemilik sumber daya, Akun AWS yang membuat ember, yang dapat mengakses ember dan objek apa pun yang dikandungnya. Namun, pemilik sumber daya dapat memilih untuk memberikan izin akses kepada sumber daya dan pengguna lain dengan menulis kebijakan akses.

Ketika Anda menetapkan kebijakan, sebaiknya Anda menyertakan string yang dihasilkan secara acak sebagai prefiks untuk bucket sehingga hanya pengaliran log yang dimaksud yang diekspor ke bucket tersebut.

> **Important**
>
> Untuk membuat ekspor ke bucket S3 lebih aman, kami sekarang meminta Anda untuk menentukan daftar akun sumber yang diizinkan untuk mengekspor data log ke bucket S3 Anda.
> Dalam contoh berikut, daftar ID akun di `aws:SourceAccount` kunci adalah akun tempat pengguna dapat mengekspor data log ke bucket S3 Anda. `aws:SourceArn`Kuncinya adalah

sumber daya tempat tindakan diambil. Anda dapat membatasi ini ke grup log tertentu, atau menggunakan wildcard seperti yang ditunjukkan dalam contoh ini.
Kami menyarankan Anda juga menyertakan ID akun akun tempat bucket S3 dibuat, untuk memungkinkan ekspor dalam akun yang sama.

Untuk mengatur izin bucket Amazon S3

1. Di konsol Amazon S3, pilih bucket yang Anda buat di langkah 1.
2. Pilih Permissions (Izin), Bucket policy (Kebijakan bucket).
3. Di Editor Kebijakan Bucket, tambahkan kebijakan berikut. Ubah `my-exported-logs` nama bucket S3 Anda. Pastikan untuk menentukan titik akhir Wilayah yang benar, seperti`us-west-1`, untuk Principal.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
              "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
         }
    ]
}
```

4. Pilih Save (Simpan) untuk menetapkan kebijakan yang baru saja ditambahkan sebagai kebijakan akses di bucket Anda. Kebijakan ini memungkinkan CloudWatch Log untuk mengekspor data log ke bucket S3 Anda. Pemilik bucket memiliki izin penuh atas semua objek yang diekspor.

> **Warning**
>
> Jika bucket yang ada sudah memiliki satu atau beberapa kebijakan yang dilampirkan padanya, tambahkan pernyataan untuk akses CloudWatch Log ke kebijakan atau kebijakan tersebut. Sebaiknya Anda mengevaluasi hasil rangkaian izin untuk memastikan bahwa itu sesuai untuk pengguna yang akan mengakses bucket.

(Opsional) Langkah 4: Mengekspor ke bucket yang dienkripsi dengan SSE-KMS

Langkah ini diperlukan hanya jika Anda mengekspor ke bucket S3 yang menggunakan enkripsi sisi server. AWS KMS keys Enkripsi ini dikenal sebagai SSE-KMS.

Untuk mengekspor ke bucket yang dienkripsi dengan SSE-KMS

1. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di bilah navigasi kiri, pilih Kunci yang dikelola pelanggan.

   Pilih Buat Kunci.
4. Untuk Tipe Kunci, pilih Simetris.
5. Untuk penggunaan Kunci, pilih Enkripsi dan dekripsi dan kemudian pilih Berikutnya.
6. Di bawah Tambahkan label, masukkan alias untuk kunci dan secara opsional tambahkan deskripsi atau tag. Lalu pilih Selanjutnya.
7. Di bawah Administrator kunci, pilih siapa yang dapat mengelola kunci ini, lalu pilih Berikutnya.
8. Di bawah Tentukan izin penggunaan kunci, jangan buat perubahan dan pilih Berikutnya.
9. Tinjau pengaturan dan pilih Selesai.
10. Kembali ke halaman kunci yang dikelola Pelanggan, pilih nama kunci yang baru saja Anda buat.
11. Pilih tab Kebijakan kunci dan pilih Beralih ke tampilan kebijakan.
12. Di bagian Kebijakan kunci, pilih Edit.
13. Tambahkan pernyataan berikut ke daftar pernyataan kebijakan kunci. Ketika Anda melakukannya, ganti *Wilayah* dengan Wilayah log Anda dan ganti *akun-ARN dengan ARN* dari akun yang memiliki kunci KMS.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
"arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
```

```
}
```

14. Pilih Simpan perubahan.
15. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
16. Temukan bucket yang Anda buat Langkah 1: Buat ember S3 dan pilih nama bucket.
17. Pilih tab Properti. Kemudian, di bawah Enkripsi Default, pilih Edit.
18. Di bawah Enkripsi sisi server, pilih Aktifkan.
19. Di bawah Tipe enkripsi memilih Kunci (SSE-KMS)AWS Key Management Service .
20. Pilih Pilih dari AWS KMS kunci Anda dan temukan kunci yang Anda buat.
21. Untuk kunci Bucket, pilih Aktifkan.
22. Pilih Simpan perubahan.

Langkah 5: Buat tugas ekspor

Di langkah ini, Anda membuat tugas ekspor untuk mengekspor log dari grup log.

Untuk mengekspor data ke Amazon S3 menggunakan konsol CloudWatch

1. Masuk dengan izin yang memadai seperti yang didokumentasikanLangkah 2: Siapkan izin akses.
2. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
3. Pada panel navigasi, pilih Grup log.
4. Di layar Log Groups (Grup Log), pilih nama grup log.
5. Pilih Actions (Tindakan), Export data to Amazon S3 (Ekspor data ke Amazon S3).
6. Di layar Export data to Amazon S3 (Ekspor data ke Amazon S3), di Define data export (Tentukan ekspor data), atur rentang waktu untuk data yang akan diekspor menggunakan From (Dari) dan To (Sampai).
7. Jika grup log Anda memiliki beberapa pengaliran log, Anda dapat memberikan prefiks pengaliran log untuk membatasi data grup log ke pengaliran tertentu. Pilih Advanced (Lanjutan), lalu untuk Stream prefix (Prefiks pengaliran), masukkan prefiks pengaliran log.
8. Di bawah bucket Pilih S3, pilih akun yang terkait dengan bucket S3.
9. Untuk nama bucket S3, pilih bucket S3.
10. Untuk S3 Bucket prefix (Prefiks bucket S3), masukkan string yang dihasilkan secara acak yang Anda tentukan dalam kebijakan bucket.

11. Pilih Export (Ekspor) untuk mengekspor data log ke Amazon S3.
12. Untuk melihat status data log yang diekspor ke Amazon S3, pilih Actions (Tindakan), lalu View all exports to Amazon S3 (Lihat semua ekspor ke Amazon S3).

Ekspor data log ke Amazon S3 menggunakan AWS CLI

Dalam contoh berikut, Anda menggunakan tugas ekspor untuk mengekspor semua data dari grup CloudWatch log Log bernama `my-log-group` ke bucket Amazon S3 bernama. `my-exported-logs` Contoh ini mengasumsikan bahwa Anda telah membuat grup log bernama `my-log-group`.

Mengekspor data log ke bucket S3 yang dienkripsi oleh didukung. AWS KMS Mengekspor ke bucket yang dienkripsi dengan DSSE-KMS tidak didukung.

Detail cara Anda mengatur ekspor tergantung pada apakah bucket Amazon S3 yang ingin Anda ekspor berada di akun yang sama dengan log Anda yang sedang diekspor, atau di akun lain.

Topik

Ekspor akun yang sama

Jika bucket Amazon S3 berada di akun yang sama dengan log yang sedang diekspor, gunakan instruksi di bagian ini.

Topik

Langkah 1: Buat ember S3

Kami menyarankan Anda menggunakan bucket yang dibuat khusus untuk CloudWatch Log. Namun, jika Anda ingin menggunakan bucket yang sudah ada, Anda dapat melompat ke langkah 2.

> **Note**
>
> Bucket S3 harus berada di Region yang sama dengan data log yang akan diekspor. CloudWatch Log tidak mendukung ekspor data ke bucket S3 di Wilayah lain.

Untuk membuat bucket S3 menggunakan AWS CLI

Di jendela perintah, jalankan perintah create-bucket berikut, di mana `LocationConstraint` adalah Wilayah tempat Anda mengekspor data log.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Berikut ini adalah output contoh.

```
{
    "Location": "/my-exported-logs"
}
```

Langkah 2: Siapkan izin akses

Untuk membuat tugas ekspor di langkah 5, Anda harus masuk dengan peran `AmazonS3ReadOnlyAccess` IAM dan dengan izin berikut:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Langkah 3: Tetapkan izin pada bucket S3

Secara default, semua bucket dan objek S3 bersifat pribadi. Hanya pemilik sumber daya, akun yang membuat bucket, yang dapat mengakses bucket dan objek yang ada di dalamnya. Namun, pemilik sumber daya dapat memilih untuk memberikan izin akses kepada sumber daya dan pengguna lain dengan menulis kebijakan akses.

> **Important**
>
> Untuk membuat ekspor ke bucket S3 lebih aman, kami sekarang meminta Anda untuk menentukan daftar akun sumber yang diizinkan untuk mengekspor data log ke bucket S3 Anda.
> Dalam contoh berikut, daftar ID akun di `aws:SourceAccount` kunci adalah akun tempat pengguna dapat mengekspor data log ke bucket S3 Anda. `aws:SourceArn`Kuncinya adalah sumber daya tempat tindakan diambil. Anda dapat membatasi ini ke grup log tertentu, atau menggunakan wildcard seperti yang ditunjukkan dalam contoh ini.
> Kami menyarankan Anda juga menyertakan ID akun akun tempat bucket S3 dibuat, untuk memungkinkan ekspor dalam akun yang sama.

Untuk menyetel izin pada bucket S3

1. Buat file bernama `policy.json` dan tambahkan kebijakan akses berikut, ubah `my-exported-logs` nama bucket S3 Anda dan `Principal` ke titik akhir Wilayah tempat Anda mengekspor data log, seperti. `us-west-1` Gunakan editor teks untuk membuat file kebijakan ini. Jangan gunakan konsol IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
```

```
                        ...
                    ]
                }
            }
        }
    ]
}
```

2. Tetapkan kebijakan yang baru saja ditambahkan sebagai kebijakan akses di bucket dengan menggunakan put-bucket-policyperintah. Kebijakan ini memungkinkan CloudWatch Log untuk mengekspor data log ke bucket S3 Anda. Pemilik bucket akan memiliki izin penuh atas semua objek yang diekspor.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> Jika bucket yang ada sudah memiliki satu atau beberapa kebijakan yang dilampirkan padanya, tambahkan pernyataan untuk akses CloudWatch Log ke kebijakan atau kebijakan tersebut. Sebaiknya Anda mengevaluasi hasil rangkaian izin untuk memastikan bahwa itu sesuai untuk pengguna yang akan mengakses bucket.

(Opsional) Langkah 4: Mengekspor ke bucket yang dienkripsi dengan SSE-KMS

Langkah ini diperlukan hanya jika Anda mengekspor ke bucket S3 yang menggunakan enkripsi sisi server. AWS KMS keys Enkripsi ini dikenal sebagai SSE-KMS.

Untuk mengekspor ke bucket yang dienkripsi dengan SSE-KMS

1. Gunakan editor teks untuk membuat file bernama `key_policy.json` dan menambahkan kebijakan akses berikut. Saat Anda menambahkan kebijakan, lakukan perubahan berikut:

   - Ganti *`Wilayah`* dengan Wilayah log Anda.
   - Ganti *`akun-ARN`* dengan ARN dari akun yang memiliki kunci KMS.

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

```
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

2. Masukkan perintah berikut:

```
aws kms create-key --policy file://key_policy.json
```

Berikut ini adalah contoh output dari perintah ini:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
```

```
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. Gunakan editor teks untuk membuat file yang disebut `bucketencryption.json` dengan konten berikut.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Masukkan perintah berikut, ganti nama *`ember dengan nama`* bucket tempat Anda mengekspor log.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

Jika perintah tidak mengembalikan kesalahan, prosesnya berhasil.

Langkah 5: Buat tugas ekspor

Gunakan perintah berikut untuk membuat tugas ekspor. Setelah Anda membuatnya, tugas ekspor mungkin memakan waktu mulai dari beberapa detik hingga beberapa jam, tergantung pada ukuran data yang akan diekspor.

Untuk mengekspor data ke Amazon S3 menggunakan AWS CLI

1. Masuk dengan izin yang memadai seperti yang didokumentasikanLangkah 2: Siapkan izin akses.
2. Pada prompt perintah, gunakan create-export-taskperintah berikut untuk membuat tugas ekspor.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-task-output"
```

Berikut ini adalah output contoh.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Ekspor lintas akun

Jika bucket Amazon S3 berada di akun yang berbeda dari log yang sedang diekspor, gunakan petunjuk di bagian ini.

Topik

- Langkah 1: Buat ember S3
- Langkah 2: Siapkan izin akses
- Langkah 3: Tetapkan izin pada bucket S3
- (Opsional) Langkah 4: Mengekspor ke bucket yang dienkripsi dengan SSE-KMS
- Langkah 5: Buat tugas ekspor

Langkah 1: Buat ember S3

Kami menyarankan Anda menggunakan bucket yang dibuat khusus untuk CloudWatch Log. Namun, jika Anda ingin menggunakan bucket yang sudah ada, Anda dapat melompat ke langkah 2.

> **Note**
>
> Bucket S3 harus berada di Region yang sama dengan data log yang akan diekspor. CloudWatch Log tidak mendukung ekspor data ke bucket S3 di Wilayah lain.

Untuk membuat bucket S3 menggunakan AWS CLI

Di jendela perintah, jalankan perintah create-bucket berikut, di mana `LocationConstraint` adalah Wilayah tempat Anda mengekspor data log.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Berikut ini adalah output contoh.

```
{
    "Location": "/my-exported-logs"
}
```

Langkah 2: Siapkan izin akses

Pertama, Anda harus membuat kebijakan IAM baru untuk mengaktifkan CloudWatch Log agar memiliki `s3:PutObject` izin untuk bucket Amazon S3 tujuan.

Untuk membuat tugas ekspor di langkah 5, Anda harus masuk dengan peran `AmazonS3ReadOnlyAccess` IAM dan dengan izin tertentu lainnya. Anda dapat membuat kebijakan yang berisi beberapa izin lain yang diperlukan ini.

Kebijakan yang Anda buat bergantung pada apakah bucket tujuan menggunakan AWS KMS enkripsi. Jika tidak menggunakan AWS KMS enkripsi, buat kebijakan dengan konten berikut.

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

```
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Jika bucket tujuan menggunakan AWS KMS enkripsi, buat kebijakan dengan konten berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

Untuk membuat tugas ekspor di langkah 5, Anda harus masuk dengan peran `AmazonS3ReadOnlyAccess` IAM, kebijakan IAM yang baru saja Anda buat, dan juga dengan izin berikut:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.

- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Langkah 3: Tetapkan izin pada bucket S3

Secara default, semua bucket dan objek S3 bersifat pribadi. Hanya pemilik sumber daya, akun yang membuat bucket, yang dapat mengakses bucket dan objek yang ada di dalamnya. Namun, pemilik sumber daya dapat memilih untuk memberikan izin akses kepada sumber daya dan pengguna lain dengan menulis kebijakan akses.

> **Important**
>
> Untuk membuat ekspor ke bucket S3 lebih aman, kami sekarang meminta Anda untuk menentukan daftar akun sumber yang diizinkan untuk mengekspor data log ke bucket S3 Anda.
> Dalam contoh berikut, daftar ID akun di `aws:SourceAccount` kunci adalah akun tempat pengguna dapat mengekspor data log ke bucket S3 Anda. `aws:SourceArn`Kuncinya adalah sumber daya tempat tindakan diambil. Anda dapat membatasi ini ke grup log tertentu, atau menggunakan wildcard seperti yang ditunjukkan dalam contoh ini.
> Kami menyarankan Anda juga menyertakan ID akun akun tempat bucket S3 dibuat, untuk memungkinkan ekspor dalam akun yang sama.

Untuk menyetel izin pada bucket S3

1. Buat file bernama `policy.json` dan tambahkan kebijakan akses berikut, ubah `my-exported-logs` nama bucket S3 Anda dan `Principal` ke titik akhir Wilayah tempat Anda mengekspor data log, seperti. `us-west-1` Gunakan editor teks untuk membuat file kebijakan ini. Jangan gunakan konsol IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
```

```
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
        }
      },
      {
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
          },
          "Action": "s3:PutObject",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control"
            }
          }
       }
    ]
}
```

2. Tetapkan kebijakan yang baru saja ditambahkan sebagai kebijakan akses di bucket dengan menggunakan put-bucket-policyperintah. Kebijakan ini memungkinkan CloudWatch Log untuk mengekspor data log ke bucket S3 Anda. Pemilik bucket akan memiliki izin penuh atas semua objek yang diekspor.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> Jika bucket yang ada sudah memiliki satu atau beberapa kebijakan yang dilampirkan padanya, tambahkan pernyataan untuk akses CloudWatch Log ke kebijakan atau

kebijakan tersebut. Sebaiknya Anda mengevaluasi hasil rangkaian izin untuk memastikan bahwa itu sesuai untuk pengguna yang akan mengakses bucket.

(Opsional) Langkah 4: Mengekspor ke bucket yang dienkripsi dengan SSE-KMS

Langkah ini diperlukan hanya jika Anda mengekspor ke bucket S3 yang menggunakan enkripsi sisi server. AWS KMS keys Enkripsi ini dikenal sebagai SSE-KMS.

Untuk mengekspor ke bucket yang dienkripsi dengan SSE-KMS

1. Gunakan editor teks untuk membuat file bernama `key_policy.json` dan menambahkan kebijakan akses berikut. Saat Anda menambahkan kebijakan, lakukan perubahan berikut:

   - Ganti *`Wilayah`* dengan Wilayah log Anda.
   - Ganti *`akun-ARN`* dengan ARN dari akun yang memiliki kunci KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
```

```
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

2. Masukkan perintah berikut:

```
aws kms create-key --policy file://key_policy.json
```

Berikut ini adalah contoh output dari perintah ini:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
```

```
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. Gunakan editor teks untuk membuat file yang disebut `bucketencryption.json` dengan konten berikut.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Masukkan perintah berikut, ganti nama *ember dengan nama* bucket tempat Anda mengekspor log.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

   Jika perintah tidak mengembalikan kesalahan, prosesnya berhasil.

Langkah 5: Buat tugas ekspor

Gunakan perintah berikut untuk membuat tugas ekspor. Setelah Anda membuatnya, tugas ekspor mungkin memakan waktu mulai dari beberapa detik hingga beberapa jam, tergantung pada ukuran data yang akan diekspor.

Untuk mengekspor data ke Amazon S3 menggunakan AWS CLI

1. Masuk dengan izin yang memadai seperti yang didokumentasikanLangkah 2: Siapkan izin akses.
2. Pada prompt perintah, gunakan create-export-taskperintah berikut untuk membuat tugas ekspor.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-
group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --
to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-
task-output"
```

Berikut ini adalah output contoh.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Jelaskan tugas ekspor

Setelah Anda membuat tugas ekspor, Anda bisa mendapatkan status tugas saat ini.

Untuk menggambarkan tugas ekspor menggunakan AWS CLI

Pada prompt perintah, gunakan describe-export-tasksperintah berikut.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id
 "cda45419-90ea-4db5-9833-aade86253e66"
```

Berikut ini adalah output contoh.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
```

```
      "tTo": 1441494000000
   }]
}
```

Anda dapat menggunakan perintah `describe-export-tasks` dalam tiga cara yang berbeda:

- Tanpa filter apa pun - Daftar semua tugas ekspor Anda, dalam urutan pembuatan terbalik.
- Filter pada ID tugas - Daftar tugas ekspor, jika ada, dengan ID yang ditentukan.
- Filter pada status tugas - Daftar tugas ekspor dengan status yang ditentukan.

Misalnya, gunakan perintah berikut untuk memfilter dengan status `FAILED`.

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

Berikut ini adalah output contoh.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

Membatalkan tugas ekspor

Anda dapat membatalkan tugas ekspor jika dalam `RUNNING` status `PENDING` atau.

Untuk membatalkan tugas ekspor menggunakan AWS CLI

Pada prompt perintah, gunakan cancel-export-taskperintah berikut:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

Anda dapat menggunakan describe-export-tasksperintah untuk memverifikasi bahwa tugas telah dibatalkan dengan sukses.

Streaming data CloudWatch Log ke OpenSearch Layanan Amazon

Anda dapat mengonfigurasi grup CloudWatch log Log untuk mengalirkan data yang diterimanya ke klaster OpenSearch Layanan Amazon Anda dalam waktu dekat melalui langganan CloudWatch Log. Untuk informasi selengkapnya, lihat Pemrosesan data log secara real-time dengan langganan.

> **Note**
>
> Streaming ke OpenSearch Layanan hanya didukung untuk grup log di kelas log Standar. Untuk informasi selengkapnya tentang kelas log, lihatKelas log.

Tergantung pada jumlah data log yang dialirkan, Anda mungkin ingin menetapkan batas eksekusi serentak tingkat fungsi pada fungsi. Untuk informasi selengkapnya, lihat Penskalaan fungsi Lambda.

> **Note**
>
> Streaming data CloudWatch Log dalam jumlah besar ke OpenSearch Layanan dapat mengakibatkan biaya penggunaan yang tinggi. Kami menyarankan Anda membuat Anggaran di AWS Billing and Cost Management konsol. Untuk informasi selengkapnya, lihat Mengelola biaya Anda dengan AWS Anggaran.

Prasyarat

Sebelum memulai, buat domain OpenSearch Layanan. Domain dapat memiliki akses publik atau akses VPC, tetapi Anda tidak dapat mengubah jenis akses setelah domain dibuat. Anda mungkin ingin meninjau setelan domain OpenSearch Layanan nanti, dan memodifikasi konfigurasi klaster berdasarkan jumlah data yang akan diproses klaster Anda. Untuk petunjuk membuat domain, lihat Membuat domain OpenSearch Layanan.

Untuk informasi selengkapnya tentang OpenSearch Layanan, lihat Panduan Pengembang OpenSearch Layanan Amazon.

Berlangganan grup log ke OpenSearch Layanan

Anda dapat menggunakan CloudWatch konsol untuk berlangganan grup log ke OpenSearch Layanan.

Untuk berlangganan grup log ke OpenSearch Layanan

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Pilih nama grup log.
4. Pilih Tindakan, Filter langganan, Buat filter langganan OpenSearch Layanan Amazon.
5. Pilih apakah Anda ingin melakukan pengaliran ke klaster di akun ini atau akun lain.
   - Jika Anda memilih akun ini, pilih domain yang Anda buat pada langkah sebelumnya.
   - Jika Anda memilih akun lain, berikan domain ARN dan endpoint.
6. Untuk Peran Eksekusi IAM Lambda, pilih peran IAM yang harus digunakan Lambda saat menjalankan panggilan. OpenSearch

   IAM role yang Anda pilih harus memenuhi persyaratan berikut:

   - Harus memiliki `lambda.amazonaws.com` dalam hubungan kepercayaan.
   - Harus mencakup kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:region:account-id:domain/target-domain-name/
*"
        }
    ]
}
```

   - Jika domain OpenSearch Layanan target menggunakan akses VPC, peran harus memiliki AWSLambdaVPCAccessExecutionRolekebijakan yang dilampirkan. Kebijakan yang dikelola

Amazon ini memberi Lambda akses ke VPC pelanggan, memungkinkan Lambda untuk menulis ke titik akhir di VPC. OpenSearch

7. Untuk format Log, pilih format log.
8. Untuk pola filter Langganan, ketikkan istilah atau pola yang akan ditemukan di peristiwa log Anda. Ini memastikan bahwa Anda hanya mengirim data yang Anda minati ke OpenSearch cluster Anda. Untuk informasi selengkapnya, lihat Membuat metrik dari peristiwa log menggunakan filter.
9. (Opsional) Untuk Pilih data log yang akan diuji, pilih aliran log lalu pilih Pola uji untuk memverifikasi bahwa filter pencarian Anda mengembalikan hasil yang Anda harapkan.
10. Pilih Mulai streaming.

Contoh kode untuk CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan CloudWatch Log dengan kit pengembangan AWS perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks pada skenario terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Contoh lintas layanan adalah contoh aplikasi yang bekerja di beberapa Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Tindakan untuk CloudWatch Log menggunakan AWS SDK
  - Gunakan AssociateKmsKey dengan AWS SDK atau CLI
  - Gunakan CancelExportTask dengan AWS SDK atau CLI
  - Gunakan CreateExportTask dengan AWS SDK atau CLI
  - Gunakan CreateLogGroup dengan AWS SDK atau CLI
  - Gunakan CreateLogStream dengan AWS SDK atau CLI
  - Gunakan DeleteLogGroup dengan AWS SDK atau CLI
  - Gunakan DeleteSubscriptionFilter dengan AWS SDK atau CLI
  - Gunakan DescribeExportTasks dengan AWS SDK atau CLI
  - Gunakan DescribeLogGroups dengan AWS SDK atau CLI
  - Gunakan DescribeSubscriptionFilters dengan AWS SDK atau CLI
  - Gunakan GetQueryResults dengan AWS SDK atau CLI
  - Gunakan PutSubscriptionFilter dengan AWS SDK atau CLI
  - Gunakan StartLiveTail dengan AWS SDK atau CLI

Tindakan untuk CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara melakukan tindakan CloudWatch Log individual dengan AWS SDK. Kutipan ini memanggil CloudWatch Logs API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi API Amazon CloudWatch Logs.

Contoh

- Gunakan AssociateKmsKey dengan AWS SDK atau CLI
- Gunakan CancelExportTask dengan AWS SDK atau CLI
- Gunakan CreateExportTask dengan AWS SDK atau CLI
- Gunakan CreateLogGroup dengan AWS SDK atau CLI
- Gunakan CreateLogStream dengan AWS SDK atau CLI
- Gunakan DeleteLogGroup dengan AWS SDK atau CLI
- Gunakan DeleteSubscriptionFilter dengan AWS SDK atau CLI
- Gunakan DescribeExportTasks dengan AWS SDK atau CLI
- Gunakan DescribeLogGroups dengan AWS SDK atau CLI
- Gunakan DescribeSubscriptionFilters dengan AWS SDK atau CLI
- Gunakan GetQueryResults dengan AWS SDK atau CLI
- Gunakan PutSubscriptionFilter dengan AWS SDK atau CLI
- Gunakan StartLiveTail dengan AWS SDK atau CLI
- Gunakan StartQuery dengan AWS SDK atau CLI

Gunakan **`AssociateKmsKey`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`AssociateKmsKey`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to associate an AWS Key Management Service (AWS KMS) key with
    /// an Amazon CloudWatch Logs log group.
    /// </summary>
    public class AssociateKmsKey
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string kmsKeyId = "arn:aws:kms:us-west-2:<account-
number>:key/7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
            string groupName = "cloudwatchlogs-example-loggroup";

            var request = new AssociateKmsKeyRequest
            {
                KmsKeyId = kmsKeyId,
                LogGroupName = groupName,
            };
```

```
            var response = await client.AssociateKmsKeyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully associated KMS key ID:
{kmsKeyId} with log group: {groupName}.");
            }
            else
            {
                Console.WriteLine("Could not make the association between:
{kmsKeyId} and {groupName}.");
            }
        }
    }
```

- Untuk detail API, lihat AssociateKmsKeydi Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`CancelExportTask`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CancelExportTask`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
using System;
using System.Threading.Tasks;
```

```
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to cancel an Amazon CloudWatch Logs export task.
/// </summary>
public class CancelExportTask
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string taskId = "exampleTaskId";

        var request = new CancelExportTaskRequest
        {
            TaskId = taskId,
        };

        var response = await client.CancelExportTaskAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"{taskId} successfully canceled.");
        }
        else
        {
            Console.WriteLine($"{taskId} could not be canceled.");
        }
    }
}
```

- Untuk detail API, lihat CancelExportTaskdi Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`CreateExportTask`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateExportTask`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Export Task to export the contents of the Amazon
    /// CloudWatch Logs to the specified Amazon Simple Storage Service (Amazon
S3)
    /// bucket.
    /// </summary>
    public class CreateExportTask
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string taskName = "export-task-example";
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string destination = "doc-example-bucket";
            var fromTime = 1437584472382;
            var toTime = 1437584472833;

            var request = new CreateExportTaskRequest
            {
```

```
                From = fromTime,
                To = toTime,
                TaskName = taskName,
                LogGroupName = logGroupName,
                Destination = destination,
            };

            var response = await client.CreateExportTaskAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"The task, {taskName} with ID: " +
                                  $"{response.TaskId} has been created
successfully.");
            }
        }
    }
```

- Untuk detail API, lihat CreateExportTaskdi Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`CreateLogGroup`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateLogGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
    using System;
```

```
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Amazon CloudWatch Logs log group.
    /// </summary>
    public class CreateLogGroup
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string logGroupName = "cloudwatchlogs-example-loggroup";

            var request = new CreateLogGroupRequest
            {
                LogGroupName = logGroupName,
            };

            var response = await client.CreateLogGroupAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully create log group with ID:
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create log group.");
            }
        }
    }
```

- Untuk detail API, lihat CreateLogGroupdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut membuat grup log bernamamy-logs:

```
aws logs create-log-group --log-group-name my-logs
```

- Untuk detail API, lihat CreateLogGroupdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import { CreateLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new CreateLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Untuk detail API, lihat CreateLogGroupdi Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`CreateLogStream`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateLogStream`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to create an Amazon CloudWatch Logs stream for a CloudWatch
/// log group.
/// </summary>
public class CreateLogStream
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string logGroupName = "cloudwatchlogs-example-loggroup";
        string logStreamName = "cloudwatchlogs-example-logstream";

        var request = new CreateLogStreamRequest
        {
            LogGroupName = logGroupName,
```

```
                LogStreamName = logStreamName,
            };

            var response = await client.CreateLogStreamAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{logStreamName} successfully created for
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create stream.");
            }
        }
    }
```

- Untuk detail API, lihat CreateLogStreamdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut membuat aliran log bernama `20150601` dalam grup log`my-logs`:

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Untuk detail API, lihat CreateLogStreamdi Referensi AWS CLI Perintah.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteLogGroup`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteLogGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Uses the Amazon CloudWatch Logs Service to delete an existing
/// CloudWatch Logs log group.
/// </summary>
public class DeleteLogGroup
{
    public static async Task Main()
    {
        var client = new AmazonCloudWatchLogsClient();
        string logGroupName = "cloudwatchlogs-example-loggroup";

        var request = new DeleteLogGroupRequest
        {
            LogGroupName = logGroupName,
        };

        var response = await client.DeleteLogGroupAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"Successfully deleted CloudWatch log group,
{logGroupName}.");
        }
    }
}
```

- Untuk detail API, lihat DeleteLogGroupdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut menghapus grup log bernamamy-logs:

```
aws logs delete-log-group --log-group-name my-logs
```

- Untuk detail API, lihat DeleteLogGroupdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import { DeleteLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Untuk detail API, lihat DeleteLogGroupdi Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteSubscriptionFilter`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteSubscriptionFilter`.

C++

SDK untuk C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Sertakan file-file yang diperlukan.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DeleteSubscriptionFilterRequest.h>
#include <iostream>
```

Hapus filter langganan.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DeleteSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetLogGroupName(log_group);

        auto outcome = cwl.DeleteSubscriptionFilter(request);
        if (!outcome.IsSuccess()) {
```

```
            std::cout << "Failed to delete CloudWatch log subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        } else {
            std::cout << "Successfully deleted CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Untuk detail API, lihat DeleteSubscriptionFilterdi Referensi AWS SDK for C++ API.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DeleteSubscriptionFilterRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <logGroup>
```

```
                Where:
                  filter - The name of the subscription filter (for example,
 MyFilter).
                  logGroup - The name of the log group. (for example, testgroup).
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String filter = args[0];
        String logGroup = args[1];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .build();

        deleteSubFilter(logs, filter, logGroup);
        logs.close();
    }

    public static void deleteSubFilter(CloudWatchLogsClient logs, String filter,
 String logGroup) {
        try {
            DeleteSubscriptionFilterRequest request =
 DeleteSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .logGroupName(logGroup)
                    .build();

            logs.deleteSubscriptionFilter(request);
            System.out.printf("Successfully deleted CloudWatch logs subscription
 filter %s", filter);

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

- Untuk detail API, lihat DeleteSubscriptionFilterdi Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import { DeleteSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteSubscriptionFilterCommand({
    // The name of the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Untuk detail API, lihat DeleteSubscriptionFilterdi Referensi AWS SDK for JavaScript API.

SDK untuk JavaScript (v2)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

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

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  filterName: "FILTER",
  logGroupName: "LOG_GROUP",
};

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

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.
- Untuk detail API, lihat DeleteSubscriptionFilterdi Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
suspend fun deleteSubFilter(
    filter: String?,
    logGroup: String?,
) {
    val request =
```

```
        DeleteSubscriptionFilterRequest {
            filterName = filter
            logGroupName = logGroup
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { logs ->
        logs.deleteSubscriptionFilter(request)
        println("Successfully deleted CloudWatch logs subscription filter named
 $filter")
    }
}
```

- Untuk detail API, lihat DeleteSubscriptionFilterdi AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DescribeExportTasks`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeExportTasks`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to retrieve a list of information about Amazon CloudWatch
    /// Logs export tasks.
```

```
    /// </summary>
    public class DescribeExportTasks
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            var request = new DescribeExportTasksRequest
            {
                Limit = 5,
            };

            var response = new DescribeExportTasksResponse();

            do
            {
                response = await client.DescribeExportTasksAsync(request);
                response.ExportTasks.ForEach(t =>
                {
                    Console.WriteLine($"{t.TaskName} with ID: {t.TaskId} has
status: {t.Status}");
                });
            }
            while (response.NextToken is not null);
        }
    }
```

- Untuk detail API, lihat DescribeExportTasksdi Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DescribeLogGroups`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeLogGroups`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Retrieves information about existing Amazon CloudWatch Logs log groups
/// and displays the information on the console.
/// </summary>
public class DescribeLogGroups
{
    public static async Task Main()
    {
        // Creates a CloudWatch Logs client using the default
        // user. If you need to work with resources in another
        // AWS Region than the one defined for the default user,
        // pass the AWS Region as a parameter to the client constructor.
        var client = new AmazonCloudWatchLogsClient();

        bool done = false;
        string newToken = null;

        var request = new DescribeLogGroupsRequest
        {
            Limit = 5,
        };

        DescribeLogGroupsResponse response;

        do
        {
            if (newToken is not null)
```

```
                {
                    request.NextToken = newToken;
                }

                response = await client.DescribeLogGroupsAsync(request);

                response.LogGroups.ForEach(lg =>
                {
                    Console.WriteLine($"{lg.LogGroupName} is associated with the
key: {lg.KmsKeyId}.");
                    Console.WriteLine($"Created on:
{lg.CreationTime.Date.Date}");
                    Console.WriteLine($"Date for this group will be stored for:
{lg.RetentionInDays} days.\n");
                });

                if (response.NextToken is null)
                {
                    done = true;
                }
                else
                {
                    newToken = response.NextToken;
                }
            }
            while (!done);
        }
    }
```

- Untuk detail API, lihat DescribeLogGroupsdi Referensi AWS SDK for .NET API.

CLI

AWS CLI

Perintah berikut menjelaskan grup log bernamamy-logs:

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

Output:

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- Untuk detail API, lihat DescribeLogGroupsdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import {
  paginateDescribeLogGroups,
  CloudWatchLogsClient,
} from "@aws-sdk/client-cloudwatch-logs";

const client = new CloudWatchLogsClient({});

export const main = async () => {
  const paginatedLogGroups = paginateDescribeLogGroups({ client }, {});
  const logGroups = [];

  for await (const page of paginatedLogGroups) {
    if (page.logGroups && page.logGroups.every((lg) => !!lg)) {
      logGroups.push(...page.logGroups);
    }
  }
```

```
  console.log(logGroups);
  return logGroups;
};
```

- Untuk detail API, lihat DescribeLogGroupsdi Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DescribeSubscriptionFilters`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeSubscriptionFilters`.

C++

SDK untuk C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Sertakan file-file yang diperlukan.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DescribeSubscriptionFiltersRequest.h>
#include <aws/logs/model/DescribeSubscriptionFiltersResult.h>
#include <iostream>
#include <iomanip>
```

Buat daftar filter berlangganan.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
```

```
        Aws::CloudWatchLogs::Model::DescribeSubscriptionFiltersRequest request;
        request.SetLogGroupName(log_group);
        request.SetLimit(1);

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = cwl.DescribeSubscriptionFilters(
                    request);
            if (!outcome.IsSuccess()) {
                std::cout << "Failed to describe CloudWatch subscription filters
"
                    << "for log group " << log_group << ": " <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(32) << "Name" <<
                    std::setw(64) << "FilterPattern" << std::setw(64) <<
                    "DestinationArn" << std::endl;
                header = true;
            }

            const auto &filters = outcome.GetResult().GetSubscriptionFilters();
            for (const auto &filter : filters) {
                std::cout << std::left << std::setw(32) <<
                    filter.GetFilterName() << std::setw(64) <<
                    filter.GetFilterPattern() << std::setw(64) <<
                    filter.GetDestinationArn() << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Untuk detail API, lihat DescribeSubscriptionFiltersdi Referensi AWS SDK for C++ API.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersResponse
import software.amazon.awssdk.services.cloudwatchlogs.model.SubscriptionFilter;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeSubscriptionFilters {
    public static void main(String[] args) {

        final String usage = """

                Usage:
                  <logGroup>

                Where:
                  logGroup - A log group name (for example, myloggroup).
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String logGroup = args[0];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        describeFilters(logs, logGroup);
        logs.close();
    }

    public static void describeFilters(CloudWatchLogsClient logs, String
logGroup) {
        try {
            boolean done = false;
            String newToken = null;

            while (!done) {
                DescribeSubscriptionFiltersResponse response;
                if (newToken == null) {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .logGroupName(logGroup)
                            .limit(1).build();

                    response = logs.describeSubscriptionFilters(request);
                } else {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .nextToken(newToken)
                            .logGroupName(logGroup)
                            .limit(1).build();
                    response = logs.describeSubscriptionFilters(request);
                }

                for (SubscriptionFilter filter : response.subscriptionFilters())
{
                    System.out.printf("Retrieved filter with name %s, " +
"pattern %s " + "and destination arn %s",
                            filter.filterName(),
                            filter.filterPattern(),
                            filter.destinationArn());
                }
```

```
                if (response.nextToken() == null) {
                    done = true;
                } else {
                    newToken = response.nextToken();
                }
            }

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

- Untuk detail API, lihat DescribeSubscriptionFiltersdi Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import { DescribeSubscriptionFiltersCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  // This will return a list of all subscription filters in your account
  // matching the log group name.
  const command = new DescribeSubscriptionFiltersCommand({
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
    limit: 1,
  });

  try {
```

```
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Untuk detail API, lihat DescribeSubscriptionFiltersdi Referensi AWS SDK for JavaScript API.

SDK untuk JavaScript (v2)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

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

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  logGroupName: "GROUP_NAME",
  limit: 5,
};

cwl.describeSubscriptionFilters(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.subscriptionFilters);
  }
});
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.

- Untuk detail API, lihat DescribeSubscriptionFiltersdi Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
suspend fun describeFilters(logGroup: String) {
    val request =
        DescribeSubscriptionFiltersRequest {
            logGroupName = logGroup
            limit = 1
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { cwlClient ->
        val response = cwlClient.describeSubscriptionFilters(request)
        response.subscriptionFilters?.forEach { filter ->
            println("Retrieved filter with name  ${filter.filterName} pattern
 ${filter.filterPattern} and destination ${filter.destinationArn}")
        }
    }
}
```

- Untuk detail API, lihat DescribeSubscriptionFiltersdi AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GetQueryResults`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetQueryResults`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Jalankan kueri besar

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
```

- Untuk detail API, lihat GetQueryResultsdi Referensi AWS SDK for JavaScript API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.
```

```
        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])
```

- Untuk detail API, lihat GetQueryResultsdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`PutSubscriptionFilter`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`PutSubscriptionFilter`.

C++

SDK untuk C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Sertakan file-file yang diperlukan.

```
#include <aws/core/Aws.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/PutSubscriptionFilterRequest.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Buat filter langganan.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::PutSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetFilterPattern(filter_pattern);
        request.SetLogGroupName(log_group);
        request.SetDestinationArn(dest_arn);
        auto outcome = cwl.PutSubscriptionFilter(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch logs subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Untuk detail API, lihat PutSubscriptionFilterdi Referensi AWS SDK for C++ API.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.PutSubscriptionFilterRequest;

/**
 * Before running this code example, you need to grant permission to CloudWatch
 * Logs the right to execute your Lambda function.
 * To perform this task, you can use this CLI command:
 *
 * aws lambda add-permission --function-name "lamda1" --statement-id "lamda1"
 * --principal "logs.us-west-2.amazonaws.com" --action "lambda:InvokeFunction"
 * --source-arn "arn:aws:logs:us-west-2:111111111111:log-group:testgroup:*"
 * --source-account "111111111111"
 *
 * Make sure you replace the function name with your function name and replace
 * '111111111111' with your account details.
 * For more information, see "Subscription Filters with AWS Lambda" in the
 * Amazon CloudWatch Logs Guide.
 *
 *
 * Also, before running this Java V2 code example,set up your development
 * environment,including your credentials.
 *
 * For more information,see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 */

public class PutSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <pattern> <logGroup> <functionArn>\s

                Where:
                  filter - A filter name (for example, myfilter).
                  pattern - A filter pattern (for example, ERROR).
```

```
                logGroup - A log group name (testgroup).
                functionArn - An AWS Lambda function ARN (for example,
arn:aws:lambda:us-west-2:111111111111:function:lambda1) .
               """;

       if (args.length != 4) {
           System.out.println(usage);
           System.exit(1);
       }

       String filter = args[0];
       String pattern = args[1];
       String logGroup = args[2];
       String functionArn = args[3];
       Region region = Region.US_WEST_2;
       CloudWatchLogsClient cwl = CloudWatchLogsClient.builder()
               .region(region)
               .build();

       putSubFilters(cwl, filter, pattern, logGroup, functionArn);
       cwl.close();
   }

   public static void putSubFilters(CloudWatchLogsClient cwl,
           String filter,
           String pattern,
           String logGroup,
           String functionArn) {

       try {
           PutSubscriptionFilterRequest request =
PutSubscriptionFilterRequest.builder()
                   .filterName(filter)
                   .filterPattern(pattern)
                   .logGroupName(logGroup)
                   .destinationArn(functionArn)
                   .build();

           cwl.putSubscriptionFilter(request);
           System.out.printf(
                   "Successfully created CloudWatch logs subscription filter
%s",
                   filter);
```

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

- Untuk detail API, lihat PutSubscriptionFilterdi Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
import { PutSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new PutSubscriptionFilterCommand({
    // An ARN of a same-account Kinesis stream, Kinesis Firehose
    // delivery stream, or Lambda function.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
SubscriptionFilters.html
    destinationArn: process.env.CLOUDWATCH_LOGS_DESTINATION_ARN,

    // A name for the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,

    // A filter pattern for subscribing to a filtered stream of log events.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
FilterAndPatternSyntax.html
    filterPattern: process.env.CLOUDWATCH_LOGS_FILTER_PATTERN,

    // The name of the log group. Messages in this group matching the filter
 pattern
    // will be sent to the destination ARN.
```

```
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Untuk detail API, lihat PutSubscriptionFilterdi Referensi AWS SDK for JavaScript API.

SDK untuk JavaScript (v2)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

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

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  destinationArn: "LAMBDA_FUNCTION_ARN",
  filterName: "FILTER_NAME",
  filterPattern: "ERROR",
  logGroupName: "LOG_GROUP",
};

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

```
  }
});
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK for JavaScript.
- Untuk detail API, lihat PutSubscriptionFilterdi Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`StartLiveTail`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`StartLiveTail`.

.NET

AWS SDK for .NET

Sertakan file-file yang diperlukan.

```
using Amazon;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

Mulai sesi Live Tail.

```
            var client = new AmazonCloudWatchLogsClient();
            var request = new StartLiveTailRequest
            {
                LogGroupIdentifiers = logGroupIdentifiers,
                LogStreamNames = logStreamNames,
                LogEventFilterPattern = filterPattern,
            };

            var response = await client.StartLiveTailAsync(request);

            // Catch if request fails
            if (response.HttpStatusCode != System.Net.HttpStatusCode.OK)
            {
```

```
                Console.WriteLine("Failed to start live tail session");
                return;
            }
```

Anda dapat menangani acara dari sesi Live Tail dengan dua cara:

```
            /* Method 1
            * 1). Asynchronously loop through the event stream
            * 2). Set a timer to dispose the stream and stop the Live Tail
session at the end.
            */
            var eventStream = response.ResponseStream;
            var task = Task.Run(() =>
            {
                foreach (var item in eventStream)
                {
                    if (item is LiveTailSessionUpdate liveTailSessionUpdate)
                    {
                        foreach (var sessionResult in
liveTailSessionUpdate.SessionResults)
                        {
                            Console.WriteLine("Message : {0}",
sessionResult.Message);
                        }
                    }
                    if (item is LiveTailSessionStart)
                    {
                        Console.WriteLine("Live Tail session started");
                    }
                    // On-stream exceptions are processed here
                    if (item is CloudWatchLogsEventStreamException)
                    {
                        Console.WriteLine($"ERROR: {item}");
                    }
                }
            });
            // Close the stream to stop the session after a timeout
            if (!task.Wait(TimeSpan.FromSeconds(10))){
                eventStream.Dispose();
                Console.WriteLine("End of line");
            }
```

```
            /* Method 2
            * 1). Add event handlers to each event variable
            * 2). Start processing the stream and wait for a timeout using
AutoResetEvent
            */
            AutoResetEvent endEvent = new AutoResetEvent(false);
            var eventStream = response.ResponseStream;
            using (eventStream) // automatically disposes the stream to stop the
session after execution finishes
            {
                eventStream.SessionStartReceived += (sender, e) =>
                {
                    Console.WriteLine("LiveTail session started");
                };
                eventStream.SessionUpdateReceived += (sender, e) =>
                {
                    foreach (LiveTailSessionLogEvent logEvent in
e.EventStreamEvent.SessionResults){
                        Console.WriteLine("Message: {0}", logEvent.Message);
                    }
                };
                // On-stream exceptions are captured here
                eventStream.ExceptionReceived += (sender, e) =>
                {
                    Console.WriteLine($"ERROR:
{e.EventStreamException.Message}");
                };

                eventStream.StartProcessing();
                // Stream events for this amount of time.
                endEvent.WaitOne(TimeSpan.FromSeconds(10));
                Console.WriteLine("End of line");
            }
```

- Untuk detail API, lihat StartLiveTaildi Referensi AWS SDK for .NET API.

Go

SDK untuk Go V2

Sertakan file-file yang diperlukan.

```
import (
 "context"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)
```

Tangani acara dari sesi Live Tail.

```
func handleEventStreamAsync(stream *cloudwatchlogs.StartLiveTailEventStream) {
 eventsChan := stream.Events()
 for {
  event := <-eventsChan
  switch e := event.(type) {
  case *types.StartLiveTailResponseStreamMemberSessionStart:
   log.Println("Received SessionStart event")
  case *types.StartLiveTailResponseStreamMemberSessionUpdate:
   for _, logEvent := range e.Value.SessionResults {
    log.Println(*logEvent.Message)
   }
  default:
   // Handle on-stream exceptions
   if err := stream.Err(); err != nil {
    log.Fatalf("Error occured during streaming: %v", err)
   } else if event == nil {
    log.Println("Stream is Closed")
    return
   } else {
    log.Fatalf("Unknown event type: %T", e)
   }
  }
 }
}
```

Mulai sesi Live Tail.

```
 cfg, err := config.LoadDefaultConfig(context.TODO())
```

```
if err != nil {
 panic("configuration error, " + err.Error())
}
client := cloudwatchlogs.NewFromConfig(cfg)

request := &cloudwatchlogs.StartLiveTailInput{
 LogGroupIdentifiers:   logGroupIdentifiers,
 LogStreamNames:        logStreamNames,
 LogEventFilterPattern: logEventFilterPattern,
}

response, err := client.StartLiveTail(context.TODO(), request)
// Handle pre-stream Exceptions
if err != nil {
 log.Fatalf("Failed to start streaming: %v", err)
}

// Start a Goroutine to handle events over stream
stream := response.GetStream()
go handleEventStreamAsync(stream)
```

Hentikan sesi Live Tail setelah periode waktu berlalu.

```
// Close the stream (which ends the session) after a timeout
time.Sleep(10 * time.Second)
stream.Close()
log.Println("Event stream closed")
```

- Untuk detail API, lihat StartLiveTaildi Referensi AWS SDK for Go API.

Java

SDK untuk Java 2.x

Sertakan file-file yang diperlukan.

```
import io.reactivex.FlowableSubscriber;
import io.reactivex.annotations.NonNull;
import org.reactivestreams.Subscription;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
```

```
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsAsyncClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionStart;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionUpdate;
import software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseHandler;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseStream;

import java.util.Date;
import java.util.List;
import java.util.concurrent.atomic.AtomicReference;
```

Tangani acara dari sesi Live Tail.

```
    private static StartLiveTailResponseHandler
 getStartLiveTailResponseStreamHandler(
            AtomicReference<Subscription> subscriptionAtomicReference) {
        return StartLiveTailResponseHandler.builder()
            .onResponse(r -> System.out.println("Received initial response"))
            .onError(throwable -> {
                CloudWatchLogsException e = (CloudWatchLogsException)
 throwable.getCause();
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            })
            .subscriber(() -> new FlowableSubscriber<>() {
                @Override
                public void onSubscribe(@NonNull Subscription s) {
                    subscriptionAtomicReference.set(s);
                    s.request(Long.MAX_VALUE);
                }

                @Override
                public void onNext(StartLiveTailResponseStream event) {
                    if (event instanceof LiveTailSessionStart) {
```

```
                        LiveTailSessionStart sessionStart =
(LiveTailSessionStart) event;
                        System.out.println(sessionStart);
                    } else if (event instanceof LiveTailSessionUpdate) {
                        LiveTailSessionUpdate sessionUpdate =
(LiveTailSessionUpdate) event;
                        List<LiveTailSessionLogEvent> logEvents =
sessionUpdate.sessionResults();
                        logEvents.forEach(e -> {
                            long timestamp = e.timestamp();
                            Date date = new Date(timestamp);
                            System.out.println("[" + date + "] " + e.message());
                        });
                    } else {
                        throw CloudWatchLogsException.builder().message("Unknown
event type").build();
                    }
                }

                @Override
                public void onError(Throwable throwable) {
                    System.out.println(throwable.getMessage());
                    System.exit(1);
                }

                @Override
                public void onComplete() {
                    System.out.println("Completed Streaming Session");
                }
            })
            .build();
    }
```

Mulai sesi Live Tail.

```
        CloudWatchLogsAsyncClient cloudWatchLogsAsyncClient =
                CloudWatchLogsAsyncClient.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();

        StartLiveTailRequest request =
                StartLiveTailRequest.builder()
```

```
                    .logGroupIdentifiers(logGroupIdentifiers)
                    .logStreamNames(logStreamNames)
                    .logEventFilterPattern(logEventFilterPattern)
                    .build();

        /* Create a reference to store the subscription */
        final AtomicReference<Subscription> subscriptionAtomicReference = new
AtomicReference<>(null);

        cloudWatchLogsAsyncClient.startLiveTail(request,
getStartLiveTailResponseStreamHandler(subscriptionAtomicReference));
```

Hentikan sesi Live Tail setelah periode waktu berlalu.

```
        /* Set a timeout for the session and cancel the subscription. This will:
         * 1). Close the stream
         * 2). Stop the Live Tail session
         */
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        if (subscriptionAtomicReference.get() != null) {
            subscriptionAtomicReference.get().cancel();
            System.out.println("Subscription to stream closed");
        }
```

- Untuk detail API, lihat StartLiveTaildi Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Sertakan file-file yang diperlukan.

```
import { CloudWatchLogsClient, StartLiveTailCommand } from "@aws-sdk/client-
cloudwatch-logs";
```

Tangani acara dari sesi Live Tail.

```
async function handleResponseAsync(response) {
    try {
      for await (const event of response.responseStream) {
        if (event.sessionStart !== undefined) {
          console.log(event.sessionStart);
        } else if (event.sessionUpdate !== undefined) {
          for (const logEvent of event.sessionUpdate.sessionResults) {
            const timestamp = logEvent.timestamp;
            const date = new Date(timestamp);
            console.log("[" + date + "] " + logEvent.message);
          }
        } else {
            console.error("Unknown event type");
        }
      }
    } catch (err) {
        // On-stream exceptions are captured here
        console.error(err)
    }
}
```

Mulai sesi Live Tail.

```
    const client = new CloudWatchLogsClient();

    const command = new StartLiveTailCommand({
        logGroupIdentifiers: logGroupIdentifiers,
        logStreamNames: logStreamNames,
        logEventFilterPattern: filterPattern
    });
    try{
        const response = await client.send(command);
        handleResponseAsync(response);
    } catch (err){
        // Pre-stream exceptions are captured here
        console.log(err);
    }
```

Hentikan sesi Live Tail setelah periode waktu berlalu.

```
    /* Set a timeout to close the client. This will stop the Live Tail session.
 */
    setTimeout(function() {
        console.log("Client timeout");
        client.destroy();
      }, 10000);
```

- Untuk detail API, lihat StartLiveTaildi Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Sertakan file-file yang diperlukan.

```
import aws.sdk.kotlin.services.cloudwatchlogs.CloudWatchLogsClient
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailRequest
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailResponseStream
import kotlinx.coroutines.flow.takeWhile
```

Mulai sesi Live Tail.

```
    val client = CloudWatchLogsClient.fromEnvironment()

    val request = StartLiveTailRequest {
        logGroupIdentifiers = logGroupIdentifiersVal
        logStreamNames = logStreamNamesVal
        logEventFilterPattern = logEventFilterPatternVal
    }

    val startTime = System.currentTimeMillis()

    try {
        client.startLiveTail(request) { response ->
            val stream = response.responseStream
            if (stream != null) {
                /* Set a timeout to unsubcribe from the flow. This will:
                * 1). Close the stream
                * 2). Stop the Live Tail session
```

```
                */
                stream.takeWhile { System.currentTimeMillis() - startTime <
 10000 }.collect { value ->
                    if (value is StartLiveTailResponseStream.SessionStart) {
                        println(value.asSessionStart())
                    } else if (value is
 StartLiveTailResponseStream.SessionUpdate) {
                        for (e in value.asSessionUpdate().sessionResults!!) {
                            println(e)
                        }
                    } else {
                        throw IllegalArgumentException("Unknown event type")
                    }
                }
            } else {
                throw IllegalArgumentException("No response stream")
            }
        }
    } catch (e: Exception) {
        println("Exception occurred during StartLiveTail: $e")
        System.exit(1)
    }
```

- Untuk detail API, lihat StartLiveTaildi AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Sertakan file-file yang diperlukan.

```
import boto3
import time
from datetime import datetime
```

Mulai sesi Live Tail.

```
    # Initialize the client
    client = boto3.client('logs')
```

```
    start_time = time.time()

    try:
        response = client.start_live_tail(
            logGroupIdentifiers=log_group_identifiers,
            logStreamNames=log_streams,
            logEventFilterPattern=filter_pattern
        )
        event_stream = response['responseStream']
        # Handle the events streamed back in the response
        for event in event_stream:
            # Set a timeout to close the stream.
            # This will end the Live Tail session.
            if (time.time() - start_time >= 10):
                event_stream.close()
                break
            # Handle when session is started
            if 'sessionStart' in event:
                session_start_event = event['sessionStart']
                print(session_start_event)
            # Handle when log event is given in a session update
            elif 'sessionUpdate' in event:
                log_events = event['sessionUpdate']['sessionResults']
                for log_event in log_events:
                    print('[{date}]
{log}'.format(date=datetime.fromtimestamp(log_event['timestamp']/1000),log=log_event['me
            else:
                # On-stream exceptions are captured here
                raise RuntimeError(str(event))
    except Exception as e:
        print(e)
```

- Untuk detail API, lihat StartLiveTaildi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`StartQuery`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`StartQuery`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Jalankan kueri besar

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
/**
 * Wrapper for the StartQueryCommand. Uses a static query string
 * for consistency.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 * @returns {Promise<{ queryId: string }>}
 */
async _startQuery([startDate, endDate], maxLogs = 10000) {
  try {
    return await this.client.send(
      new StartQueryCommand({
        logGroupNames: this.logGroupNames,
        queryString: "fields @timestamp, @message | sort @timestamp asc",
        startTime: startDate.valueOf(),
        endTime: endDate.valueOf(),
        limit: maxLogs,
      }),
    );
  } catch (err) {
    /** @type {string} */
    const message = err.message;
    if (message.startsWith("Query's end date and time")) {
      // This error indicates that the query's start or end date occur
      // before the log group was created.
      throw new DateOutOfBoundsError(message);
    }
```

```
      throw err;
    }
  }
```

- Untuk detail API, lihat StartQuerydi Referensi AWS SDK for JavaScript API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

```
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
```

```
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
```

```
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")
```

- Untuk detail API, lihat StartQuerydi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menerapkan skenario umum di CloudWatch Log dengan AWS SDK. Skenario ini menunjukkan cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam CloudWatch Log. Setiap skenario menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode.

Contoh

- Gunakan CloudWatch Log untuk menjalankan kueri besar

Gunakan CloudWatch Log untuk menjalankan kueri besar

Contoh kode berikut menunjukkan cara menggunakan CloudWatch Log untuk menanyakan lebih dari 10.000 catatan.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

Ini adalah titik masuknya.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { CloudWatchLogsClient } from "@aws-sdk/client-cloudwatch-logs";
import { CloudWatchQuery } from "./cloud-watch-query.js";

console.log("Starting a recursive query...");

if (!process.env.QUERY_START_DATE || !process.env.QUERY_END_DATE) {
  throw new Error(
    "QUERY_START_DATE and QUERY_END_DATE environment variables are required.",
  );
}

const cloudWatchQuery = new CloudWatchQuery(new CloudWatchLogsClient({}), {
  logGroupNames: ["/workflows/cloudwatch-logs/large-query"],
  dateRange: [
    new Date(parseInt(process.env.QUERY_START_DATE)),
    new Date(parseInt(process.env.QUERY_END_DATE)),
  ],
});

await cloudWatchQuery.run();

console.log(
  `Queries finished in ${cloudWatchQuery.secondsElapsed} seconds.\nTotal logs
 found: ${cloudWatchQuery.results.length}`,
);
```

Ini adalah kelas yang membagi kueri menjadi beberapa langkah jika perlu.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  StartQueryCommand,
  GetQueryResultsCommand,
} from "@aws-sdk/client-cloudwatch-logs";
import { splitDateRange } from "@aws-doc-sdk-examples/lib/utils/util-date.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

class DateOutOfBoundsError extends Error {}

export class CloudWatchQuery {
  /**
   * Run a query for all CloudWatch Logs within a certain date range.
   * CloudWatch logs return a max of 10,000 results. This class
   * performs a binary search across all of the logs in the provided
   * date range if a query returns the maximum number of results.
   *
   * @param {import('@aws-sdk/client-cloudwatch-logs').CloudWatchLogsClient}
 client
   * @param {{ logGroupNames: string[], dateRange: [Date, Date], queryConfig:
 { limit: number } }} config
   */
  constructor(client, { logGroupNames, dateRange, queryConfig }) {
    this.client = client;
    /**
     * All log groups are queried.
     */
    this.logGroupNames = logGroupNames;

    /**
     * The inclusive date range that is queried.
     */
    this.dateRange = dateRange;

    /**
     * CloudWatch Logs never returns more than 10,000 logs.
     */
    this.limit = queryConfig?.limit ?? 10000;

    /**
     * @type {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]}
     */
```

```
    this.results = [];
  }

  /**
   * Run the query.
   */
  async run() {
    this.secondsElapsed = 0;
    const start = new Date();
    this.results = await this._largeQuery(this.dateRange);
    const end = new Date();
    this.secondsElapsed = (end - start) / 1000;
    return this.results;
  }

  /**
   * Recursively query for logs.
   * @param {[Date, Date]} dateRange
   * @returns {Promise<import("@aws-sdk/client-cloudwatch-logs").ResultField[]
[]>}
   */
  async _largeQuery(dateRange) {
    const logs = await this._query(dateRange, this.limit);

    console.log(
      `Query date range: ${dateRange
        .map((d) => d.toISOString())
        .join(" to ")}. Found ${logs.length} logs.`,
    );

    if (logs.length < this.limit) {
      return logs;
    }

    const lastLogDate = this._getLastLogDate(logs);
    const offsetLastLogDate = new Date(lastLogDate);
    offsetLastLogDate.setMilliseconds(lastLogDate.getMilliseconds() + 1);
    const subDateRange = [offsetLastLogDate, dateRange[1]];
    const [r1, r2] = splitDateRange(subDateRange);
    const results = await Promise.all([
      this._largeQuery(r1),
      this._largeQuery(r2),
    ]);
    return [logs, ...results].flat();
```

```
}

/**
 * Find the most recent log in a list of logs.
 * @param {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]} logs
 */
_getLastLogDate(logs) {
  const timestamps = logs
    .map(
      (log) =>
        log.find((fieldMeta) => fieldMeta.field === "@timestamp")?.value,
    )
    .filter((t) => !!t)
    .map((t) => `${t}Z`)
    .sort();

  if (!timestamps.length) {
    throw new Error("No timestamp found in logs.");
  }

  return new Date(timestamps[timestamps.length - 1]);
}

// snippet-start:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
// snippet-end:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]

/**
 * Starts a query and waits for it to complete.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 */
async _query(dateRange, maxLogs) {
  try {
    const { queryId } = await this._startQuery(dateRange, maxLogs);
    const { results } = await this._waitUntilQueryDone(queryId);
    return results ?? [];
  } catch (err) {
```

```
      /**
       * This error is thrown when StartQuery returns an error indicating
       * that the query's start or end date occur before the log group was
       * created.
       */
      if (err instanceof DateOutOfBoundsError) {
        return [];
      } else {
        throw err;
      }
    }
  }

  // snippet-start:[javascript.v3.cloudwatch-logs.actions.StartQuery]
  /**
   * Wrapper for the StartQueryCommand. Uses a static query string
   * for consistency.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   * @returns {Promise<{ queryId: string }>}
   */
  async _startQuery([startDate, endDate], maxLogs = 10000) {
    try {
      return await this.client.send(
        new StartQueryCommand({
          logGroupNames: this.logGroupNames,
          queryString: "fields @timestamp, @message | sort @timestamp asc",
          startTime: startDate.valueOf(),
          endTime: endDate.valueOf(),
          limit: maxLogs,
        }),
      );
    } catch (err) {
      /** @type {string} */
      const message = err.message;
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }

      throw err;
    }
  }
```

```
  // snippet-end:[javascript.v3.cloudwatch-logs.actions.StartQuery]

  /**
   * Call GetQueryResultsCommand until the query is done.
   * @param {string} queryId
   */
  _waitUntilQueryDone(queryId) {
    const getResults = async () => {
      const results = await this._getQueryResults(queryId);
      const queryDone = [
        "Complete",
        "Failed",
        "Cancelled",
        "Timeout",
        "Unknown",
      ].includes(results.status);

      return { queryDone, results };
    };

    return retry(
      { intervalInMs: 1000, maxRetries: 60, quiet: true },
      async () => {
        const { queryDone, results } = await getResults();
        if (!queryDone) {
          throw new Error("Query not done.");
        }

        return results;
      },
    );
  }
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for JavaScript .
  - GetQueryResults
  - StartQuery

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di AWS Repositori Contoh Kode.

File ini memanggil modul contoh untuk mengelola CloudWatch kueri melebihi 10.000 hasil.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import os
import sys

import boto3
from botocore.config import Config

from cloudwatch_query import CloudWatchQuery
from date_utilities import DateUtilities

# Configure logging at the module level.
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(filename)s:%(lineno)d - %(message)s",
)


class CloudWatchLogsQueryRunner:
    def __init__(self):
        """
        Initializes the CloudWatchLogsQueryRunner class by setting up date
 utilities
        and creating a CloudWatch Logs client with retry configuration.
        """
        self.date_utilities = DateUtilities()
        self.cloudwatch_logs_client = self.create_cloudwatch_logs_client()

    def create_cloudwatch_logs_client(self):
        """
```

```
        Creates and returns a CloudWatch Logs client with a specified retry
configuration.

        :return: A CloudWatch Logs client instance.
        :rtype: boto3.client
        """
        try:
            return boto3.client("logs", config=Config(retries={"max_attempts":
10}))
        except Exception as e:
            logging.error(f"Failed to create CloudWatch Logs client: {e}")
            sys.exit(1)

    def fetch_environment_variables(self):
        """
        Fetches and validates required environment variables for query start and
end dates.

        :return: Tuple of query start date and end date as integers.
        :rtype: tuple
        :raises SystemExit: If required environment variables are missing or
invalid.
        """
        try:
            query_start_date = int(os.environ["QUERY_START_DATE"])
            query_end_date = int(os.environ["QUERY_END_DATE"])
        except KeyError:
            logging.error(
                "Both QUERY_START_DATE and QUERY_END_DATE environment variables
are required."
            )
            sys.exit(1)
        except ValueError as e:
            logging.error(f"Error parsing date environment variables: {e}")
            sys.exit(1)

        return query_start_date, query_end_date

    def convert_dates_to_iso8601(self, start_date, end_date):
        """
        Converts UNIX timestamp dates to ISO 8601 format using DateUtilities.

        :param start_date: The start date in UNIX timestamp.
        :type start_date: int
```

```
        :param end_date: The end date in UNIX timestamp.
        :type end_date: int
        :return: Start and end dates in ISO 8601 format.
        :rtype: tuple
        """
        start_date_iso8601 =
 self.date_utilities.convert_unix_timestamp_to_iso8601(
            start_date
        )
        end_date_iso8601 = self.date_utilities.convert_unix_timestamp_to_iso8601(
            end_date
        )
        return start_date_iso8601, end_date_iso8601

    def execute_query(
        self,
        start_date_iso8601,
        end_date_iso8601,
        log_group="/workflows/cloudwatch-logs/large-query",
    ):
        """
        Creates a CloudWatchQuery instance and executes the query with provided
 date range.

        :param start_date_iso8601: The start date in ISO 8601 format.
        :type start_date_iso8601: str
        :param end_date_iso8601: The end date in ISO 8601 format.
        :type end_date_iso8601: str
        :param log_group: Log group to search: "/workflows/cloudwatch-logs/large-
query"
        :type log_group: str
        """
        cloudwatch_query = CloudWatchQuery(
            [start_date_iso8601, end_date_iso8601],
        )
        cloudwatch_query.query_logs((start_date_iso8601, end_date_iso8601))
        logging.info("Query executed successfully.")
        logging.info(
            f"Queries completed in {cloudwatch_query.query_duration} seconds.
 Total logs found: {len(cloudwatch_query.query_results)}"
        )


def main():
```

```
    """
    Main function to start a recursive CloudWatch logs query.
    Fetches required environment variables, converts dates, and executes the
 query.
    """
    logging.info("Starting a recursive CloudWatch logs query...")
    runner = CloudWatchLogsQueryRunner()
    query_start_date, query_end_date = runner.fetch_environment_variables()
    start_date_iso8601 = DateUtilities.convert_unix_timestamp_to_iso8601(
        query_start_date
    )
    end_date_iso8601 =
 DateUtilities.convert_unix_timestamp_to_iso8601(query_end_date)
    runner.execute_query(start_date_iso8601, end_date_iso8601)


if __name__ == "__main__":
    main()
```

Modul ini memproses CloudWatch kueri melebihi 10.000 hasil.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import time
from datetime import datetime
import threading
import boto3

from date_utilities import DateUtilities


class DateOutOfBoundsError(Exception):
    """Exception raised when the date range for a query is out of bounds."""

    pass


class CloudWatchQuery:
    """
    A class to query AWS CloudWatch logs within a specified date range.
```

```
    :ivar date_range: Start and end datetime for the query.
    :vartype date_range: tuple
    :ivar limit: Maximum number of log entries to return.
    :vartype limit: int
    """

    def __init__(self, date_range):
        self.lock = threading.Lock()
        self.log_groups = "/workflows/cloudwatch-logs/large-query"
        self.query_results = []
        self.date_range = date_range
        self.query_duration = None
        self.datetime_format = "%Y-%m-%d %H:%M:%S.%f"
        self.date_utilities = DateUtilities()
        self.limit = 10000

    def query_logs(self, date_range):
        """
        Executes a CloudWatch logs query for a specified date range and
calculates the execution time of the query.

        :return: A batch of logs retrieved from the CloudWatch logs query.
        :rtype: list
        """
        start_time = datetime.now()

        start_date, end_date = self.date_utilities.normalize_date_range_format(
            date_range, from_format="unix_timestamp", to_format="datetime"
        )

        logging.info(
            f"Original query:"
            f"\n       START:    {start_date}"
            f"\n       END:      {end_date}"
        )
        self.recursive_query((start_date, end_date))
        end_time = datetime.now()
        self.query_duration = (end_time - start_time).total_seconds()

    def recursive_query(self, date_range):
        """
        Processes logs within a given date range, fetching batches of logs
recursively if necessary.
```

```
        :param date_range: The date range to fetch logs for, specified as a tuple
(start_timestamp, end_timestamp).
        :type date_range: tuple
        :return: None if the recursive fetching is continued or stops when the
final batch of logs is processed.
                 Although it doesn't explicitly return the query results, this
method accumulates all fetched logs
                 in the `self.query_results` attribute.
        :rtype: None
        """
        batch_of_logs = self.perform_query(date_range)
        # Add the batch to the accumulated logs
        with self.lock:
            self.query_results.extend(batch_of_logs)
        if len(batch_of_logs) == self.limit:
            logging.info(f"Fetched {self.limit}, checking for more...")
            most_recent_log = self.find_most_recent_log(batch_of_logs)
            most_recent_log_timestamp = next(
                item["value"]
                for item in most_recent_log
                if item["field"] == "@timestamp"
            )
            new_range = (most_recent_log_timestamp, date_range[1])
            midpoint = self.date_utilities.find_middle_time(new_range)

            first_half_thread = threading.Thread(
                target=self.recursive_query,
                args=((most_recent_log_timestamp, midpoint),),
            )
            second_half_thread = threading.Thread(
                target=self.recursive_query, args=((midpoint, date_range[1]),)
            )

            first_half_thread.start()
            second_half_thread.start()

            first_half_thread.join()
            second_half_thread.join()

    def find_most_recent_log(self, logs):
        """
        Search a list of log items and return most recent log entry.
        :param logs: A list of logs to analyze.
        :return: log
```

```
        :type :return List containing log item details
        """
        most_recent_log = None
        most_recent_date = "1970-01-01 00:00:00.000"

        for log in logs:
            for item in log:
                if item["field"] == "@timestamp":
                    logging.debug(f"Compared: {item['value']} to
{most_recent_date}")
                    if (
                        self.date_utilities.compare_dates(
                            item["value"], most_recent_date
                        )
                        == item["value"]
                    ):
                        logging.debug(f"New most recent: {item['value']}")
                        most_recent_date = item["value"]
                        most_recent_log = log
        logging.info(f"Most recent log date of batch: {most_recent_date}")
        return most_recent_log

    # snippet-start:[python.example_code.cloudwatch_logs.start_query]
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
```

```
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(
```

```
self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")

    # snippet-end:[python.example_code.cloudwatch_logs.start_query]

    # snippet-start:[python.example_code.cloudwatch_logs.get_query_results]
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])

    # snippet-end:[python.example_code.cloudwatch_logs.get_query_results]
```

- Untuk detail API, lihat topik berikut ini adalah Referensi API SDK untuk Python (Boto3)AWS .

- GetQueryResults
- StartQuery

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh lintas layanan untuk CloudWatch Log menggunakan AWS SDK

Contoh aplikasi berikut menggunakan AWS SDK untuk menggabungkan CloudWatch Log dengan lainnya Layanan AWS. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan aplikasi.

Contoh

- Menggunakan peristiwa terjadwal untuk menginvokasi fungsi Lambda

Menggunakan peristiwa terjadwal untuk menginvokasi fungsi Lambda

Contoh kode berikut menunjukkan cara membuat AWS Lambda fungsi yang dipanggil oleh acara EventBridge terjadwal Amazon.

Python

SDK untuk Python (Boto3)

Contoh ini menunjukkan cara mendaftarkan AWS Lambda fungsi sebagai target EventBridge acara Amazon terjadwal. Penangan Lambda menulis pesan ramah dan data peristiwa lengkap ke Amazon CloudWatch Logs untuk pengambilan nanti.

- Menyebarkan fungsi Lambda.
- Membuat acara EventBridge terjadwal dan menjadikan fungsi Lambda sebagai target.
- Memberikan izin untuk membiarkan EventBridge menjalankan fungsi Lambda.
- Mencetak data terbaru dari CloudWatch Log untuk menampilkan hasil pemanggilan terjadwal.
- Membersihkan semua sumber daya yang dibuat selama demo.

Contoh ini paling baik dilihat di GitHub. Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- CloudWatch Log
- EventBridge
- Lambda

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan CloudWatch Log dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan di Amazon CloudWatch Log

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

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

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

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon CloudWatch Logs. Ini menunjukkan kepada Anda cara mengonfigurasi CloudWatch Log Amazon untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya CloudWatch Log Anda.

Konten

Perlindungan data di Amazon CloudWatch Logs

> **Note**
>
> Selain informasi berikut tentang perlindungan data umum di AWS, CloudWatch Log juga memungkinkan Anda untuk melindungi data sensitif dalam peristiwa log dengan menutupinya. Untuk informasi selengkapnya, lihat Membantu melindungi data log sensitif dengan masking.

Model tanggung jawab AWS bersama model tanggung berlaku untuk perlindungan data di Amazon CloudWatch Logs. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

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

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

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

Enkripsi diam

CloudWatch Log melindungi data saat istirahat menggunakan enkripsi. Semua grup log dienkripsi. Secara default, layanan CloudWatch Log mengelola kunci enkripsi sisi server.

Jika Anda ingin mengelola kunci yang digunakan untuk mengenkripsi dan mendekripsi log Anda, gunakan kunci. AWS KMS Untuk informasi selengkapnya, lihat Enkripsi data log di CloudWatch Log menggunakan AWS Key Management Service.

Enkripsi bergerak

CloudWatch Log menggunakan end-to-end enkripsi data dalam perjalanan. Layanan CloudWatch Log mengelola kunci enkripsi sisi server.

Manajemen identitas dan akses untuk Amazon CloudWatch Logs

Akses ke Amazon CloudWatch Logs memerlukan kredensyal yang AWS dapat digunakan untuk mengautentikasi permintaan Anda. Kredensyal tersebut harus memiliki izin untuk mengakses AWS sumber daya, seperti untuk mengambil data CloudWatch Log tentang sumber daya cloud Anda. Bagian berikut memberikan rincian tentang bagaimana Anda dapat menggunakan AWS Identity and Access Management (IAM) dan CloudWatch Log untuk membantu mengamankan sumber daya Anda dengan mengontrol siapa yang dapat mengaksesnya:

- Autentikasi
- Kontrol akses

Autentikasi

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .
- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Kontrol akses

Anda dapat memiliki kredensyal yang valid untuk mengautentikasi permintaan Anda, tetapi kecuali Anda memiliki izin, Anda tidak dapat membuat atau mengakses sumber daya Log. CloudWatch Misalnya, Anda harus memiliki izin untuk membuat pengaliran log, membuat grup log, dan sebagainya.

Bagian berikut menjelaskan cara mengelola izin untuk CloudWatch Log. Anda sebaiknya membaca gambaran umum terlebih dahulu.

- Ikhtisar mengelola izin akses ke sumber daya CloudWatch Log Anda
- Menggunakan kebijakan berbasis identitas (kebijakan IAM) untuk Log CloudWatch
- CloudWatch Referensi izin log

Ikhtisar mengelola izin akses ke sumber daya CloudWatch Log Anda

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Topik

- CloudWatch Log sumber daya dan operasi
- Memahami kepemilikan sumber daya
- Mengelola akses ke sumber daya
- Menentukan elemen kebijakan: Tindakan, efek, dan penanggung jawab
- Menetapkan ketentuan dalam kebijakan

CloudWatch Log sumber daya dan operasi

Di CloudWatch Log, sumber daya utama adalah grup log, aliran log, dan tujuan. CloudWatch Log tidak mendukung subsumber daya (sumber daya lain untuk digunakan dengan sumber daya utama).

Sumber daya dan sub-sumber daya ini memiliki nama Amazon Resource Name (ARN) yang unik seperti yang ditunjukkan pada tabel berikut.

Jenis sumber daya	Format ARN
Grup log	Kedua hal berikut ini digunakan. Yang kedua, dengan `:*` di akhir, adalah apa yang dikembalikan oleh perintah `describe-log-groups` CLI dan API. DescribeLogGroups arn:aws:logs:*region*:*account-id* :log-group:*log_group_name*

Jenis sumber daya	Format ARN
	*arn:aws:logs: wilayah: account-id:log-group: log_group_name : ** Gunakan versi pertama, tanpa trailing`:*`, dalam situasi berikut: • Di bidang `logGroupIdentifier` input di banyak CloudWatch Logs API. • Di `resourceArn` bidang dalam menandai API • Dalam IAM kebijakan, saat menentukan izin untuk TagResource, UntagResource, dan. ListTagsForResource Gunakan versi kedua, dengan tambahan`:*`, untuk merujuk ke ARN saat menentukan izin dalam kebijakan IAM untuk semua tindakan API lainnya.
Pengaliran log	*arn:aws:logs: wilayah: account-id:log-group: log_group_name:log-stream: log-stream-name*
Tujuan	arn:aws:logs:*region*:*account-id* :destination:*destination_name*

Untuk informasi selengkapnya tentang ARN, lihat ARN dalam Panduan Pengguna IAM. Untuk informasi tentang ARN CloudWatch Log, lihat Nama Sumber Daya Amazon (ARN) di. Referensi Umum Amazon Web Services Untuk contoh kebijakan yang mencakup CloudWatch Log, lihatMenggunakan kebijakan berbasis identitas (kebijakan IAM) untuk Log CloudWatch .

CloudWatch Log menyediakan serangkaian operasi untuk bekerja dengan sumber daya CloudWatch Log. Untuk daftar operasi yang tersedia, lihat CloudWatch Referensi izin log.

Memahami kepemilikan sumber daya

AWS Akun memiliki sumber daya yang dibuat di akun, terlepas dari siapa yang membuat sumber daya. Secara khusus, pemilik sumber daya adalah AWS akun entitas utama (yaitu, akun root, pengguna, atau peran IAM) yang mengautentikasi permintaan pembuatan sumber daya. Contoh berikut menggambarkan cara kerjanya:

- Jika Anda menggunakan kredensyal akun root AWS akun Anda untuk membuat grup log, AWS akun Anda adalah pemilik sumber daya CloudWatch Log.
- Jika Anda membuat pengguna di AWS akun Anda dan memberikan izin untuk membuat sumber daya CloudWatch Log kepada pengguna tersebut, pengguna dapat membuat sumber daya CloudWatch Log. Namun, AWS akun Anda, yang menjadi milik pengguna, memiliki sumber daya CloudWatch Log.
- Jika Anda membuat peran IAM di AWS akun Anda dengan izin untuk membuat sumber daya CloudWatch Log, siapa pun yang dapat mengambil peran tersebut dapat membuat sumber daya CloudWatch Log. AWS Akun Anda, tempat peran tersebut berada, memiliki sumber daya CloudWatch Log.

Mengelola akses ke sumber daya

Kebijakan izin menjelaskan siapa yang memiliki akses ke suatu objek. Bagian berikut menjelaskan opsi yang tersedia untuk membuat kebijakan izin.

> **Note**
>
> Bagian ini membahas penggunaan IAM dalam konteks Log. CloudWatch Bagian ini tidak memberikan informasi yang mendetail tentang layanan IAM. Untuk dokumentasi lengkap IAM, lihat Apa yang Dimaksud dengan IAM? dalam Panduan Pengguna IAM. Untuk informasi tentang sintaksis dan penjelasan kebijakan IAM, lihat Referensi Kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan yang melekat pada identitas IAM disebut sebagai kebijakan berbasis identitas (kebijakan IAM) dan kebijakan yang melekat pada sumber daya disebut sebagai kebijakan berbasis sumber daya. CloudWatch Log mendukung kebijakan berbasis identitas, dan kebijakan berbasis sumber daya untuk tujuan, yang digunakan untuk mengaktifkan langganan lintas akun. Untuk informasi selengkapnya, lihat Langganan lintas akun Lintas wilayah.

Topik

Izin grup log dan Wawasan Kontributor

Contributor Insights adalah fitur CloudWatch yang memungkinkan Anda menganalisis data dari grup log dan membuat deret waktu yang menampilkan data kontributor. Anda dapat melihat metrik tentang kontributor N teratas, total kontributor unik, dan penggunaannya. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Kontributor untuk Menganalisis Data Berkardinalitas Tinggi.

Saat Anda memberikan izin `cloudwatch:PutInsightRule` dan `cloudwatch:GetInsightRuleReport` izin kepada pengguna, pengguna tersebut dapat membuat aturan yang mengevaluasi grup log apa pun di CloudWatch Log dan kemudian melihat hasilnya. Hasil dapat memuat data kontributor untuk grup log tersebut. Pastikan untuk memberikan izin ini hanya kepada pengguna yang harus dapat melihat data ini.

Kebijakan berbasis sumber daya

CloudWatch Log mendukung kebijakan berbasis sumber daya untuk tujuan, yang dapat Anda gunakan untuk mengaktifkan langganan lintas akun. Untuk informasi selengkapnya, lihat Langkah 1: Buat tujuan. Tujuan dapat dibuat menggunakan PutDestinationAPI, dan Anda dapat menambahkan kebijakan sumber daya ke tujuan menggunakan PutDestinationPolicyAPI. Contoh berikut memungkinkan AWS akun lain dengan ID akun 111122223333 untuk berlangganan grup log mereka ke tujuan. `arn:aws:logs:us-east-1:123456789012:destination:testDestination`

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-east-1:123456789012:destination:testDestination"
    }
  ]
}
```

Menentukan elemen kebijakan: Tindakan, efek, dan penanggung jawab

Untuk setiap sumber daya CloudWatch Log, layanan mendefinisikan satu set operasi API. Untuk memberikan izin untuk operasi API ini, CloudWatch Log mendefinisikan serangkaian tindakan yang dapat Anda tentukan dalam kebijakan. Beberapa operasi API dapat memerlukan izin untuk lebih dari satu tindakan untuk melakukan operasi API. Untuk informasi selengkapnya tentang sumber daya dan operasi API, lihat CloudWatch Log sumber daya dan operasi dan CloudWatch Referensi izin log.

Berikut ini adalah elemen-elemen kebijakan dasar:

- Sumber daya – Anda menggunakan Amazon Resource Name (ARN) untuk mengidentifikasi sumber daya yang diberlakukan oleh kebijakan tersebut. Untuk informasi selengkapnya, lihat CloudWatch Log sumber daya dan operasi.
- Tindakan – Anda menggunakan kata kunci tindakan untuk mengidentifikasi operasi sumber daya yang ingin Anda izinkan atau tolak. Misalnya, izin `logs.DescribeLogGroups` memungkinkan pengguna untuk melakukan `DescribeLogGroups` operasi.
- Pengaruh – Anda menetapkan pengaruh, baik memperbolehkan atau menolak, ketika pengguna meminta tindakan tertentu. Jika Anda tidak secara eksplisit memberikan akses ke (mengizinkan) sumber daya, akses akan ditolak secara implisit. Anda juga dapat secara eksplisit menolak akses ke sumber daya, yang mungkin Anda lakukan untuk memastikan bahwa pengguna tidak dapat mengaksesnya, meskipun kebijakan yang berbeda memberikan akses.
- Principal – Dalam kebijakan berbasis identitas (Kebijakan IAM), pengguna yang kebijakannya terlampir adalah principal yang implisit. Untuk kebijakan berbasis sumber daya, Anda menentukan pengguna, akun, layanan, atau entitas lain yang ingin Anda terima izin (hanya berlaku untuk kebijakan berbasis sumber daya). CloudWatch Log mendukung kebijakan berbasis sumber daya untuk tujuan.

Untuk mempelajari selengkapnya tentang sintaksis dan deskripsi kebijakan IAM, lihat Referensi Kebijakan IAM AWS dalam Panduan Pengguna IAM.

Untuk tabel yang menampilkan semua tindakan API CloudWatch Log dan sumber daya yang diterapkan, lihatCloudWatch Referensi izin log.

Menetapkan ketentuan dalam kebijakan

Ketika Anda memberikan izin, Anda dapat menggunakan bahasa kebijakan akses untuk menentukan syarat ketika kebijakan akan berlaku. Misalnya, Anda mungkin ingin kebijakan diterapkan hanya

setelah tanggal tertentu. Untuk informasi selengkapnya tentang menentukan kondisi dalam bahasa kebijakan, lihat Kondisi dalam Panduan Pengguna IAM.

Untuk menyatakan kondisi, Anda menggunakan kunci kondisi standar. Untuk daftar kunci konteks yang didukung oleh setiap AWS layanan dan daftar kunci kebijakan AWS-wide, lihat Kunci tindakan, sumber daya, dan kondisi untuk AWS layanan dan kunci konteks kondisi AWS global.

> **Note**
>
> Anda dapat menggunakan tag untuk mengontrol akses ke sumber CloudWatch Log, termasuk grup log dan tujuan. Akses ke aliran log dikontrol pada tingkat grup log, karena hubungan hierarkis antara grup log dan aliran log. Untuk informasi selengkapnya tentang penggunaan tanda untuk mengendalikan akses, lihat Mengendalikan akses ke sumber daya Amazon Web Services menggunakan tanda.

Menggunakan kebijakan berbasis identitas (kebijakan IAM) untuk Log CloudWatch

Topik ini memberikan contoh kebijakan berbasis identitas di mana administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran).

> **Important**
>
> Kami menyarankan Anda terlebih dahulu meninjau topik pengantar yang menjelaskan konsep dasar dan opsi yang tersedia bagi Anda untuk mengelola akses ke sumber daya CloudWatch Log Anda. Untuk informasi selengkapnya, lihat Ikhtisar mengelola izin akses ke sumber daya CloudWatch Log Anda.

Topik ini mencakup hal-hal berikut:

- Izin yang diperlukan untuk menggunakan konsol CloudWatch
- AWS kebijakan terkelola (standar) untuk CloudWatch Log
- Contoh kebijakan yang dikelola pelanggan

Berikut ini adalah contoh kebijakan izin:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Kebijakan ini memiliki satu pernyataan yang memberikan izin untuk membuat grup log dan pengaliran log, untuk mengunggah log acara ke pengaliran log, dan daftar detail tentang pengaliran log.

Karakter wildcard (*) di akhir nilai `Resource` berarti bahwa pernyataan memungkinkan izin untuk tindakan `logs:CreateLogGroup`, `logs:CreateLogStream`, `logs:PutLogEvents`, dan `logs:DescribeLogStreams` di setiap grup log. Untuk membatasi izin ini ke grup log tertentu, ganti karakter wildcard (*) di ARN sumber daya dengan ARN grup log tertentu. Untuk informasi selengkapnya tentang bagian-bagian dalam pernyataan kebijakan IAM, lihat Referensi Elemen Kebijakan IAM dalam Panduan Pengguna IAM. Untuk daftar yang menampilkan semua tindakan CloudWatch Log, lihatCloudWatch Referensi izin log.

Izin yang diperlukan untuk menggunakan konsol CloudWatch

Agar pengguna dapat bekerja dengan CloudWatch Log di CloudWatch konsol, pengguna tersebut harus memiliki seperangkat izin minimum yang memungkinkan pengguna mendeskripsikan AWS sumber daya lain di AWS akun mereka. Untuk menggunakan CloudWatch Log di CloudWatch konsol, Anda harus memiliki izin dari layanan berikut:

- CloudWatch
- CloudWatch Log
- OpenSearch Layanan
- IAM

- Kinesis
- Lambda
- Amazon S3

Jika Anda membuat kebijakan IAM yang lebih ketat dari izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksudkan untuk pengguna dengan kebijakan IAM tersebut. Untuk memastikan bahwa pengguna tersebut masih dapat menggunakan CloudWatch konsol, lampirkan juga kebijakan `CloudWatchReadOnlyAccess` terkelola ke pengguna, seperti yang dijelaskan dalamAWS kebijakan terkelola (standar) untuk CloudWatch Log.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau API CloudWatch Log.

Set lengkap izin yang diperlukan untuk bekerja dengan CloudWatch konsol untuk pengguna yang tidak menggunakan konsol untuk mengelola langganan log adalah:

- jam tangan awan: GetMetricData
- jam tangan awan: ListMetrics
- log: CancelExportTask
- log: CreateExportTask
- log: CreateLogGroup
- log: CreateLogStream
- log: DeleteLogGroup
- log: DeleteLogStream
- log: DeleteMetricFilter
- log: DeleteQueryDefinition
- log: DeleteRetentionPolicy
- log: DeleteSubscriptionFilter
- log: DescribeExportTasks
- log: DescribeLogGroups
- log: DescribeLogStreams
- log: DescribeMetricFilters
- log: DescribeQueryDefinitions
- log: DescribeQueries

- log: DescribeSubscriptionFilters
- log: FilterLogEvents
- log: GetLogEvents
- log: GetLogGroupFields
- log: GetLogRecord
- log: GetQueryResults
- log: PutMetricFilter
- log: PutQueryDefinition
- log: PutRetentionPolicy
- log: StartQuery
- log: StopQuery
- log: PutSubscriptionFilter
- log: TestMetricFilter

Untuk pengguna yang juga akan menggunakan konsol untuk mengelola langganan log, izin berikut juga diperlukan:

- es: DescribeElasticsearchDomain
- es: ListDomainNames
- saya: AttachRolePolicy
- saya: CreateRole
- saya: GetPolicy
- saya: GetPolicyVersion
- saya: GetRole
- saya: ListAttachedRolePolicies
- saya: ListRoles
- kinesis: DescribeStreams
- kinesis: ListStreams
- lambda: AddPermission
- lambda: CreateFunction
- lambda: GetFunctionConfiguration

- lambda: ListAliases
- lambda: ListFunctions
- lambda: ListVersionsByFunction
- lambda: RemovePermission
- s3: ListBuckets

AWS kebijakan terkelola (standar) untuk CloudWatch Log

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh. AWS Kebijakan terkelola memberikan izin yang diperlukan untuk kasus penggunaan umum sehingga Anda tidak perlu menyelidiki izin apa yang diperlukan. Untuk informasi selengkapnya, lihat Kebijakan Terkelola AWS dalam Panduan Pengguna IAM.

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna dan peran di akun Anda, khusus untuk CloudWatch Log:

- CloudWatchLogsFullAccess— Memberikan akses penuh ke CloudWatch Log.
- CloudWatchLogsReadOnlyAccess— Memberikan akses hanya-baca ke Log. CloudWatch

CloudWatchLogsFullAccess

CloudWatchLogsFullAccessKebijakan ini memberikan akses penuh ke CloudWatch Log. Kebijakan tersebut menyertakan `cloudwatch:GenerateQuery` izin, sehingga pengguna dengan kebijakan ini dapat menghasilkan string kueri Wawasan CloudWatch Log dari prompt bahasa alami. Isinya sebagai berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
```

```
}
```

CloudWatchLogsReadOnlyAccess

CloudWatchLogsReadOnlyAccessKebijakan ini memberikan akses hanya-baca ke Log. CloudWatch Ini menyertakan `cloudwatch:GenerateQuery` izin, sehingga pengguna dengan kebijakan ini dapat menghasilkan string kueri Wawasan CloudWatch Log dari prompt bahasa alami. Isinya sebagai berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatchLogsCrossAccountSharingConfiguration

CloudWatchLogsCrossAccountSharingConfigurationKebijakan ini memberikan akses untuk membuat, mengelola, dan melihat tautan Pengelola Akses Observabilitas untuk berbagi sumber CloudWatch Log antar akun. Untuk informasi lebih lanjut, lihat CloudWatch observabilitas lintas akun.

Isinya sebagai berikut:

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

```
        {
            "Effect": "Allow",
            "Action": [
                "logs:Link",
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

CloudWatch Log pembaruan ke kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk CloudWatch Log sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen CloudWatch Log.

Perubahan	Deskripsi	Tanggal
		27 November 2023

Perubahan	Deskripsi	Tanggal
CloudWatchLogsFullAccess— Perbarui ke kebijakan yang ada.	CloudWatch Log menambahkan izin ke CloudWatchLogsFullAccess. `cloudwatch:GenerateQuery` Izin ditambahkan, sehingga pengguna dengan kebijakan ini dapat menghasilkan string kueri Wawasan CloudWatch Log dari prompt bahasa alami.	
CloudWatchLogsReadOnlyAccess— Perbarui ke kebijakan yang ada.	CloudWatch menambahkan izin untuk CloudWatchLogsReadOnlyAccess. `cloudwatch:GenerateQuery` Izin ditambahkan, sehingga pengguna dengan kebijakan ini dapat menghasilkan string kueri Wawasan CloudWatch Log dari prompt bahasa alami.	27 November 2023

Perubahan	Deskripsi	Tanggal
CloudWatchLogsReadOnlyAccess – Pembaruan ke kebijakan yang ada	CloudWatch Log menambahkan izin ke CloudWatchLogsReadOnlyAccess. Izin `logs:StartLiveTail` dan `logs:StopLiveTail` izin ditambahkan sehingga pengguna dengan kebijakan ini dapat menggunakan konsol untuk memulai dan menghentikan sesi ekor langsung CloudWatch Log. Untuk informasi selengkapnya, silakan lihat Menggunakan live tail untuk melihat log mendekati waktu nyata.	6 Juni 2023
CloudWatchLogsCrossAccountSharingConfiguration – Kebijakan baru	CloudWatch Log menambahkan kebijakan baru untuk memungkinkan Anda mengelola tautan pengamatan CloudWatch lintas akun yang berbagi grup CloudWatch log Log. Untuk informasi lebih lanjut, lihat CloudWatch observabilitas lintas akun	27 November 2022

Perubahan	Deskripsi	Tanggal
CloudWatchLogsReadOnlyAccess – Pembaruan ke kebijakan yang ada	CloudWatch Log menambahkan izin ke CloudWatchLogsReadOnlyAccess. Izin `oam:ListSinks` dan `oam:ListAttachedLinks` izin ditambahkan sehingga pengguna dengan kebijakan ini dapat menggunakan konsol untuk melihat data yang dibagikan dari akun sumber dalam pengamatan CloudWatch lintas akun.	27 November 2022

Contoh kebijakan yang dikelola pelanggan

Anda dapat membuat kebijakan IAM kustom Anda sendiri untuk mengizinkan izin untuk tindakan dan sumber CloudWatch daya Log. Anda dapat menyematkan kebijakan khusus ini untuk pengguna atau grup yang memerlukan izin tersebut.

Di bagian ini, Anda dapat menemukan contoh kebijakan pengguna yang memberikan izin untuk berbagai tindakan CloudWatch Log. Kebijakan ini berfungsi saat Anda menggunakan CloudWatch Logs API, AWS SDK, atau file. AWS CLI

Contoh

- Contoh 1: Izinkan akses penuh ke CloudWatch Log
- Contoh 2: Izinkan akses hanya-baca ke Log CloudWatch
- Contoh 3: Izinkan akses ke satu grup log

Contoh 1: Izinkan akses penuh ke CloudWatch Log

Kebijakan berikut memungkinkan pengguna mengakses semua tindakan CloudWatch Log.

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

Contoh 2: Izinkan akses hanya-baca ke Log CloudWatch

AWS menyediakan CloudWatchLogsReadOnlyAccesskebijakan yang memungkinkan akses hanya-baca ke data CloudWatch Log. Kebijakan ini mencakup izin berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Contoh 3: Izinkan akses ke satu grup log

Kebijakan berikut mengizinkan pengguna untuk membaca dan menulis log acara dalam satu grup log tertentu.

> **Important**
>
> `:*`Di akhir nama grup log di `Resource` baris diperlukan untuk menunjukkan bahwa kebijakan berlaku untuk semua aliran log di grup log ini. Jika Anda menghilangkan`:*`, kebijakan tidak akan diberlakukan.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

Menggunakan penandaan dan kebijakan IAM untuk mengendalikan di tingkat grup log

Anda dapat memberi pengguna akses ke grup log tertentu serta mencegah mereka mengakses grup log lainnya. Untuk melakukannya, beri tanda grup log Anda dan gunakan kebijakan IAM yang merujuk ke tanda tersebut. Untuk menerapkan tag ke grup log, Anda harus memiliki `logs:TagLogGroup` izin `logs:TagResource` atau izin. Ini berlaku baik jika Anda menetapkan tag ke grup log saat Anda membuatnya. atau menetapkannya nanti.

Untuk informasi selengkapnya tentang penandaan grup log, lihat Tandai grup log di Amazon CloudWatch Logs.

Ketika Anda menandai grup log, Anda kemudian dapat memberikan kebijakan IAM kepada pengguna untuk mengizinkan akses hanya ke grup log dengan tanda tertentu. Sebagai contoh, pernyataan

kebijakan berikut ini memberikan akses ke hanya grup log dengan nilai `Green` untuk kunci tanda `Team`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

Operasi StopLiveTailAPI StopQuerydan tidak berinteraksi dengan AWS sumber daya dalam pengertian tradisional. Mereka tidak mengembalikan data apa pun, memasukkan data apa pun, atau memodifikasi sumber daya dengan cara apa pun. Sebaliknya, mereka hanya beroperasi pada sesi ekor langsung tertentu atau kueri Wawasan CloudWatch Log tertentu, yang tidak dikategorikan sebagai sumber daya. Akibatnya, ketika Anda menentukan `Resource` bidang dalam kebijakan IAM untuk operasi ini, Anda harus menetapkan nilai `Resource` bidang sebagai*, seperti pada contoh berikut.

```
{
    "Version": "2012-10-17",
    "Statement":
        [ {
            "Effect": "Allow",
            "Action": [
                "logs:StopQuery",
                "logs:StopLiveTail"
            ],
            "Resource": "*"
            }
        ]
```

```
}
```

Untuk informasi selengkapnya tentang menggunakan pernyataan kebijakan IAM, lihat Mengendalikan Akses Menggunakan Kebijakan dalam Panduan Pengguna IAM.

CloudWatch Referensi izin log

Ketika Anda mengatur Kontrol akses dan menulis kebijakan izin yang dapat Anda lampirkan ke identitas IAM (kebijakan berbasis identitas), Anda dapat menggunakan tabel berikut sebagai referensi. Tabel mencantumkan setiap operasi API CloudWatch Log dan tindakan terkait yang dapat Anda berikan izin untuk melakukan tindakan. Anda menentukan tindakan di bidang `Action` kebijakan. Untuk `Resource` bidang, Anda dapat menentukan ARN grup log atau aliran log, atau menentukan `*` untuk mewakili semua sumber CloudWatch Log.

Anda dapat menggunakan kunci kondisi AWS-wide dalam kebijakan CloudWatch Log Anda untuk menyatakan kondisi. Untuk daftar lengkap kunci AWS-wide, lihat Kunci Konteks Kondisi AWS Global dan IAM di Panduan Pengguna IAM.

> **Note**
>
> Untuk menentukan tindakan, gunakan awalan `logs:` diikuti dengan nama operasi API. Misalnya:`logs:CreateLogGroup`,`logs:CreateLogStream`, atau `logs:*` (untuk semua tindakan CloudWatch Log).

CloudWatch Log operasi API dan izin yang diperlukan untuk tindakan

CloudWatch Log operasi API	Izin yang diperlukan (tindakan API)
CancelExportTask	`logs:CancelExportTask` Diperlukan untuk membatalkan tugas ekspor yang tertunda atau berjalan.
CreateExportTask	`logs:CreateExportTask` Diperlukan untuk mengekspor data dari grup log ke buket Amazon S3.
CreateLogGroup	`logs:CreateLogGroup`

CloudWatch Log operasi API	Izin yang diperlukan (tindakan API)
	Diperlukan untuk membuat grup log baru.
CreateLogStream	`logs:CreateLogStream` Diperlukan untuk membuat aliran log baru dalam grup log.
DeleteDestination	`logs:DeleteDestination` Diperlukan untuk menghapus tujuan log dan menonaktifkan penyaring berlangganan apa pun.
DeleteLogGroup	`logs:DeleteLogGroup` Diperlukan untuk menghapus grup log dan semua peristiwa log yang diarsipkan yang terkait.
DeleteLogStream	`logs:DeleteLogStream` Diperlukan untuk menghapus aliran log dan peristiwa log yang diarsipkan yang terkait.
DeleteMetricFilter	`logs:DeleteMetricFilter` Diperlukan untuk menghapus penyaring metrik yang terkait dengan grup log.
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Diperlukan untuk menghapus definisi kueri yang disimpan di Wawasan CloudWatch Log.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Diperlukan untuk menghapus kebijakan sumber daya CloudWatch Log.

CloudWatch Log operasi API	Izin yang diperlukan (tindakan API)
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Diperlukan untuk menghapus kebijakan penyimpanan grup log.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Diperlukan untuk menghapus penyaring berlangganan yang terkait dengan grup log.
DescribeDestinations	`logs:DescribeDestinations` Diperlukan untuk melihat semua destinasi yang terkait dengan akun.
DescribeExportTasks	`logs:DescribeExportTasks` Diperlukan untuk melihat semua tugas ekspor yang terkait dengan akun.
DescribeLogGroups	`logs:DescribeLogGroups` Diperlukan untuk melihat semua grup log yang terkait dengan akun.
DescribeLogStreams	`logs:DescribeLogStreams` Diperlukan untuk melihat semua aliran log yang terkait dengan grup log.
DescribeMetricFilters	`logs:DescribeMetricFilters` Diperlukan untuk melihat semua metrik yang terkait dengan grup log.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Diperlukan untuk melihat daftar definisi kueri yang disimpan di Wawasan CloudWatch Log.

CloudWatch Log operasi API	Izin yang diperlukan (tindakan API)
DescribeQueries	`logs:DescribeQueries` Diperlukan untuk melihat daftar kueri Wawasan CloudWatch Log yang dijadwalkan, dijalankan, atau baru-baru ini dikeluarkan.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Diperlukan untuk melihat daftar kebijakan sumber daya CloudWatch Log.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Diperlukan untuk melihat semua penyaring berlangganan yang terkait dengan grup log.
FilterLogEvents	`logs:FilterLogEvents` Diperlukan untuk mengurutkan peristiwa log berdasarkan pola penyaringan grup log.
GetLogEvents	`logs:GetLogEvents` Diperlukan untuk mengambil kejadian log dari aliran log.
GetLogGroupFields	`logs:GetLogGroupFields` Diperlukan untuk mengambil daftar kolom yang disertakan dalam peristiwa log di grup log.
GetLogRecord	`logs:GetLogRecord` Diperlukan untuk mengambil rincian dari satu peristiwa log.

CloudWatch Log operasi API	Izin yang diperlukan (tindakan API)
GetQueryResults	`logs:GetQueryResults` Diperlukan untuk mengambil hasil kueri Wawasan CloudWatch Log.
ListTagsLogGroup	`logs:ListTagsLogGroup` Diperlukan untuk membuat daftar tag yang terkait dengan grup log.
PutDestination	`logs:PutDestination` Diperlukan untuk membuat atau memperbarui aliran log tujuan (seperti aliran Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Diperlukan untuk membuat atau memperbarui kebijakan akses yang terkait dengan tujuan log yang sudah ada.
PutLogEvents	`logs:PutLogEvents` Diperlukan untuk mengunggah kumpulan peristiwa log ke aliran log.
PutMetricFilter	`logs:PutMetricFilter` Diperlukan untuk membuat atau memperbarui penyaring metrik dan mengaitkannya dengan grup log.
PutQueryDefinition	`logs:PutQueryDefinition` Diperlukan untuk menyimpan kueri di Wawasan CloudWatch Log.

CloudWatch Log operasi API	Izin yang diperlukan (tindakan API)
PutResourcePolicy	`logs:PutResourcePolicy` Diperlukan untuk membuat kebijakan sumber daya CloudWatch Log.
PutRetentionPolicy	`logs:PutRetentionPolicy` Diperlukan untuk mengatur jumlah hari untuk menyimpan peristiwa (penyimpanan) log dalam grup log.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Diperlukan untuk membuat atau memperbarui penyaring berlangganan dan mengaitkannya dengan grup log.
StartQuery	`logs:StartQuery` Diperlukan untuk memulai kueri Wawasan CloudWatch Log.
StopQuery	`logs:StopQuery` Diperlukan untuk menghentikan kueri Wawasan CloudWatch Log yang sedang berlangsung.
TagLogGroup	`logs:TagLogGroup` Perlu menambahkan atau memperbarui tag grup log.
TestMetricFilter	`logs:TestMetricFilter` Diperlukan untuk menguji pola penyaringan terhadap sampel pesan peristiwa log.

Menggunakan peran terkait layanan untuk Log CloudWatch

Amazon CloudWatch Logs menggunakan AWS Identity and Access Management peran terkait layanan (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke Log. CloudWatch Peran terkait layanan telah ditentukan sebelumnya oleh CloudWatch Log dan menyertakan semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan CloudWatch Log lebih efisien karena Anda tidak diharuskan menambahkan izin yang diperlukan secara manual. CloudWatch Log mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya CloudWatch Log yang dapat mengambil peran tersebut. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin. Kebijakan izin itu tidak dapat dilampirkan ke entitas IAM lainnya.

Untuk informasi tentang layanan lain yang mendukung peran yang terhubung dengan layanan, lihat AWS Layanan yang Bekerja dengan IAM. Cari layanan yang memiliki Yes di kolom Service-Linked Role. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk Log CloudWatch

CloudWatch Log menggunakan nama peran terkait layanan. AWSServiceRoleForLogDelivery CloudWatch Log menggunakan peran terkait layanan ini untuk menulis log langsung ke Firehose. Untuk informasi selengkapnya, lihat Aktifkan pencatatan dari AWS layanan.

Peran tertaut layanan AWSServiceRoleForLogDelivery memercayai layanan berikut untuk mengambil peran tersebut:

- `logs.amazonaws.com`

Kebijakan izin peran memungkinkan CloudWatch Log untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `firehose:PutRecord` dan `firehose:PutRecordBatch` pada semua aliran Firehose yang memiliki tag dengan `LogDeliveryEnabled` kunci dengan nilai. `True` Tag ini secara otomatis dilampirkan ke aliran Firehose saat Anda membuat langganan untuk mengirimkan log ke Firehose.

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM untuk membuat, mengedit, atau menghapus peran yang terhubung dengan layanan. Entitas ini dapat berupa pengguna, grup, atau peran. Untuk informasi lebih lanjut, lihat Izin Peran yang Terhubung dengan Layanan di Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Log CloudWatch

Anda tidak perlu membuat peran yang terhubung dengan layanan secara manual. Saat Anda menyiapkan log untuk dikirim langsung ke aliran Firehose di AWS Management Console, the, atau AWS API AWS CLI, CloudWatch Log akan membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda mengatur kembali log untuk dikirim langsung ke aliran Firehose, CloudWatch Log akan membuat peran terkait layanan untuk Anda lagi.

Mengedit peran terkait layanan untuk Log CloudWatch

CloudWatch Log tidak memungkinkan Anda untuk mengedit AWSServiceRoleForLogDelivery, atau peran terkait layanan lainnya, setelah Anda membuatnya. Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit Peran Tertaut Layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Log CloudWatch

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran yang terhubung dengan layanan sebelum menghapusnya secara manual.

> **Note**
>
> Jika layanan CloudWatch Log menggunakan peran saat Anda mencoba menghapus sumber daya, penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus sumber daya CloudWatch Log yang digunakan oleh AWSServiceRoleForLogDeliveryperan terkait layanan

- Berhenti mengirim log langsung ke aliran Firehose.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran AWSServiceRoleForLogDeliveryterkait layanan. Untuk informasi selengkapnya, lihat Menghapus Peran yang Terhubung dengan Layanan

Wilayah yang Didukung untuk CloudWatch peran terkait layanan Log

CloudWatch Log mendukung penggunaan peran terkait layanan di semua AWS Wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat CloudWatch Logs Regions and Endpoints.

Validasi kepatuhan untuk Amazon Logs CloudWatch

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon CloudWatch Logs sebagai bagian dari beberapa program AWS kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat AWS Layanan dalam Lingkup olehAWS Layanan Program Kepatuhan . Untuk informasi umum, lihat Program KepatuhanAWS .

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

Tanggung jawab kepatuhan Anda saat menggunakan Amazon CloudWatch Logs ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah–langkah untuk melakukan deployment terhadap lingkungan dasar di AWSyang menjadi fokus keamanan dan kepatuhan.
- Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang sesuai dengan HIPAA.

- AWS Sumber DayaAWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Mengevaluasi Sumber Daya dengan Aturan dalam PanduanAWS Config Pengembang — AWS Config; menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Ketahanan di Amazon CloudWatch Logs

Infrastruktur global AWS dibangun berdasarkan Wilayah AWS dan Availability Zone. Wilayah menyediakan beberapa Availability Zone yang terpisah dan terisolasi secara fisik, yang terhubung melalui jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Availability Zone, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami failover antar zona tanpa gangguan. Availability Zone lebih tersedia, memiliki toleransi kesalahan, dan dapat diskalakan dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.

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

Keamanan infrastruktur di Amazon CloudWatch Logs

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

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

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

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

Menggunakan CloudWatch Log dengan titik akhir VPC antarmuka

Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meng-host AWS sumber daya Anda, Anda dapat membuat koneksi pribadi antara VPC dan Log Anda. CloudWatch Anda dapat menggunakan koneksi ini untuk mengirim CloudWatch log ke Log tanpa mengirimnya melalui internet.

Amazon VPC adalah AWS layanan yang dapat Anda gunakan untuk meluncurkan AWS sumber daya di jaringan virtual yang Anda tentukan. Dengan VPC, Anda memiliki kendali terhadap pengaturan jaringan, seperti rentang alamat IP, subnet, tabel rute, dan pintu masuk jaringan. Untuk menghubungkan VPC Anda ke CloudWatch Log, Anda menentukan titik akhir VPC antarmuka untuk Log. CloudWatch Jenis titik akhir ini memungkinkan Anda untuk menghubungkan VPC Anda ke layanan AWS . Endpoint menyediakan konektivitas yang andal dan dapat diskalakan ke CloudWatch Log tanpa memerlukan gateway internet, instance terjemahan alamat jaringan (NAT), atau koneksi VPN. Untuk informasi selengkapnya, lihat Apa yang dimaksud dengan Amazon VPC dalam Panduan Pengguna Amazon VPC.

Endpoint VPC antarmuka didukung oleh AWS PrivateLink, sebuah AWS teknologi yang memungkinkan komunikasi pribadi antara AWS layanan menggunakan antarmuka jaringan elastis dengan alamat IP pribadi. Untuk informasi selengkapnya, lihat Baru — AWS PrivateLink untuk AWS Layanan.

Langkah-langkah berikut ditujukan untuk para pengguna Amazon VPC. Untuk informasi selengkapnya, silakan lihat Getting Started di Panduan Pengguna Amazon VPC.

Ketersediaan

CloudWatch Log saat ini mendukung titik akhir VPC di semua AWS Wilayah, termasuk Wilayah. AWS GovCloud (US)

Membuat titik akhir VPC untuk Log CloudWatch

Untuk mulai menggunakan CloudWatch Log dengan VPC Anda, buat antarmuka VPC endpoint untuk Log. CloudWatch Layanan yang harus dipilih adalah com.amazonaws.***Region***.logs. Anda tidak perlu

mengubah pengaturan apa pun untuk CloudWatch Log. Untuk informasi selengkapnya, silakan lihat Membuat sebuah Titik Akhir Antarmuka dalam Panduan Pengguna Amazon VPC.

Menguji koneksi antara VPC dan Log CloudWatch

Setelah Anda membuat titik akhir, Anda dapat menguji koneksi.

Untuk menguji koneksi antara VPC dan titik akhir Log CloudWatch

1. Connect ke instans Amazon EC2 yang berada di VPC Anda. Untuk informasi tentang menghubungkan, lihat Hubungkan ke Instans Linux Anda atau Connect ke Instans Windows Andadalam dokumentasi Amazon EC2.
2. Dari contoh, gunakan AWS CLI untuk membuat entri log di salah satu grup log yang ada.

   Pertama, buat file JSON dengan log acara. Stempel waktu harus ditetapkan sebagai angka dalam milidetik setelah 1 Jan 1970 00:00:00 UTC.

   ```
   [
     {
       "timestamp": 1533854071310,
       "message": "VPC Connection Test"
     }
   ]
   ```

   Kemudian, gunakan perintah `put-log-events` untuk membuat entri log:

   ```
   aws logs put-log-events --log-group-name LogGroupName --log-stream-
   name LogStreamName --log-events file://JSONFileName
   ```

   Jika respons terhadap perintah termasuk `nextSequenceToken`, perintah telah berhasil dan VPC endpoint Anda bekerja.

Mengontrol akses ke titik akhir VPC CloudWatch Log

Kebijakan titik akhir VPC adalah kebijakan sumber daya IAM yang Anda lampirkan ke titik akhir ketika membuat atau mengubah titik akhir. Jika Anda tidak melampirkan kebijakan ketika membuat titik akhir, kami melampirkan kebijakan default untuk Anda sehingga memungkinkan akses penuh ke layanan. Kebijakan endpoint tidak mengesampingkan atau mengganti kebijakan 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, silakan lihat Mengendalikan Akses ke Layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Berikut ini adalah contoh kebijakan endpoint untuk CloudWatch Log. Kebijakan ini memungkinkan pengguna yang terhubung ke CloudWatch Log melalui VPC untuk membuat aliran log dan mengirim CloudWatch log ke Log, serta mencegah mereka melakukan tindakan Log lainnya CloudWatch .

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Untuk mengubah kebijakan titik akhir VPC untuk Log CloudWatch

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Titik Akhir.
3. Jika Anda belum membuat endpoint untuk CloudWatch Log, pilih Create Endpoint. Kemudian pilih com.amazonaws.***Region***.logs dan pilih Create endpoint (Buat titik akhir).
4. Pilih titik akhir com.amazonaws.***Region***.logs, dan pilih tab Policy (Kebijakan) di bagian bawah layar.
5. Pilih Edit Policy (Edit Kebijakan) dan buat perubahan pada kebijakan.

Support untuk kunci konteks VPC

CloudWatch Log mendukung `aws:SourceVpc` dan kunci `aws:SourceVpce` konteks yang dapat membatasi akses ke VPC tertentu atau titik akhir VPC tertentu. Kunci ini bekerja hanya ketika pengguna menggunakan VPC endpoint. Untuk informasi selengkapnya, lihat Kunci yang Tersedia untuk Beberapa Layanan di Panduan Pengguna IAM.

Logging CloudWatch Logs API dan operasi konsol di AWS CloudTrail

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

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

Topik

CloudWatch Informasi log di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas peristiwa yang didukung terjadi di CloudWatch Log, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Peristiwa. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk peristiwa untuk CloudWatch Log, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan

lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

CloudWatch Log mendukung pencatatan tindakan berikut sebagai peristiwa dalam file CloudTrail log:

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

Hanya elemen permintaan yang masuk CloudTrail untuk tindakan API CloudWatch Log ini:

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- FilterLogEvents
- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults

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

- Apakah permintaan tersebut dibuat dengan kredensial root atau pengguna IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen CloudTrail UserIdentity.

Informasi pembuatan kueri di CloudTrail

CloudTrail logging untuk acara konsol generator Query juga didukung. Generator kueri saat ini didukung untuk Wawasan CloudWatch Log dan Wawasan CloudWatch Metrik. Dalam CloudTrail peristiwa ini, `eventSource` adalah`monitoring.amazonaws.com`.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan GenerateQuerytindakan di Wawasan CloudWatch Log.

```
{
```

```
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId": "123456789012",
                "userName": "SAMPLE_NAME"
            },
            "attributes": {
                "creationDate": "2020-04-08T21:43:24Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "monitoring.amazonaws.com",
    "eventName": "GenerateQuery",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "exampleUserAgent",
    "requestParameters": {
        "query_ask": "***",
        "query_type": "LogsInsights",
        "logs_insights": {
            "fields": "***",
            "log_group_names": ["yourloggroup"]
        },
        "include_description": true
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
```

```
}
```

Memahami entri file log

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

Entri file log berikut menunjukkan bahwa pengguna bernama CreateExportTasktindakan CloudWatch Log.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
```

```
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

CloudWatch Referensi agen log

> **Important**
>
> Referensi ini untuk agen Log lama yang tidak digunakan lagi. CloudWatch Jika Anda menggunakan Instance Metadata Service Version 2 (IMDSv2), Anda harus menggunakan agen terpadu yang baru. CloudWatch Bahkan jika Anda tidak menggunakan IMDSv2, kami sangat menyarankan Anda menggunakan CloudWatch agen terpadu yang lebih baru daripada agen log yang lebih lama. Untuk informasi selengkapnya tentang agen terpadu yang lebih baru, lihat Mengumpulkan metrik dan log dari instans Amazon EC2 dan server lokal dengan agen. CloudWatch
> Untuk informasi tentang migrasi dari agen CloudWatch Log lama ke agen terpadu, lihat Membuat file konfigurasi CloudWatch agen dengan wizard.

Agen CloudWatch Log menyediakan cara otomatis untuk mengirim data log ke Log dari CloudWatch instans Amazon EC2. Agen meliputi komponen berikut:

- Plug-in untuk AWS CLI yang mendorong data log ke CloudWatch Log.
- Skrip (daemon) yang memulai proses untuk mendorong data ke Log. CloudWatch
- Tugas cron yang memastikan bahwa daemon selalu berjalan.

File konfigurasi agen

File konfigurasi agen CloudWatch Log menjelaskan informasi yang dibutuhkan oleh agen CloudWatch Log. Bagian [general] file konfigurasi agen mendefinisikan konfigurasi umum yang berlaku untuk semua pengaliran log. Bagian [logstream] menentukan informasi yang diperlukan untuk mengirim file lokal ke pengaliran log jarak jauh. Anda dapat memiliki lebih dari satu bagian [logstream], tetapi masing-masing harus memiliki nama yang unik dalam file konfigurasi, misalnya [logstream1], [logstream2], dan seterusnya. Nilai [logstream] beserta baris pertama data dalam berkas log akan menentukan identitas berkas log.

```
[general]
state_file = value
logging_config_file = value
use_gzip_http_content_encoding = [true | false]
```

```
[logstream1]
log_group_name = value
log_stream_name = value
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

state_file

Menentukan tempat file state disimpan.

logging_config_file

(Opsional) Menentukan lokasi file konfigurasi pencatatan agen. Jika Anda tidak menentukan file konfigurasi pencatatan agen di sini, file default awslogs.conf akan digunakan. Lokasi file default adalah `/var/awslogs/etc/awslogs.conf` jika Anda menginstal agen dengan skrip, dan `/etc/awslogs/awslogs.conf` jika Anda menginstal agen dengan rpm. File ini dalam format file konfigurasi Python (https://docs.python.org/2/library/logging.config.html #logging-config-fileformat). Pencatat log dengan nama berikut dapat disesuaikan.

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

Contoh di bawah ini mengubah tingkat pembaca dan penerbit menjadi WARNING sementara nilai default-nya adalah INFO.

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

Saat disetel ke true (default), aktifkan pengkodean konten http gzip untuk mengirim muatan terkompresi ke Log. CloudWatch Ini mengurangi penggunaan CPU, menurunkan NetworkOut, dan mengurangi latensi put. Untuk menonaktifkan fitur ini, tambahkan use_gzip_http_content_encoding = false ke bagian [umum] dari file konfigurasi agen Logs, lalu restart agen. CloudWatch

> **Note**
>
> Pengaturan ini hanya tersedia di awscli-cwlogs versi 1.3.3 dan yang lebih baru.

log_group_name

Menentukan grup log tujuan. Jika belum ada, grup log akan dibuat secara otomatis. Nama grup log dapat berisi antara 1 dan 512 karakter. Karakter yang diperbolehkan meliputi a–z, A–Z, 0–9, '_' (garis bawah), '-' (tanda hubung), '/' (garis miring), dan '.' (titik).

log_stream_name

Menentukan pengaliran log tujuan. Anda dapat menggunakan string literal atau variabel yang telah ditetapkan ({instance_id}, {hostname}, {ip_address}), atau kombinasi keduanya untuk menentukan nama pengaliran log. Jika belum ada, pengaliran log akan dibuat secara otomatis.

datetime_format

Menentukan bagaimana stempel waktu diekstraksi dari log. Stempel waktu digunakan untuk mengambil log acara dan menghasilkan metrik. Waktu saat ini akan digunakan untuk setiap log acara jika datetime_format tidak disediakan. Jika nilai datetime_format yang diberikan tidak valid untuk pesan log tertentu, stempel waktu dari log acara terakhir dengan stempel waktu yang berhasil diurai akan digunakan. Jika tidak ada log acara sebelumnya, waktu saat ini akan digunakan.

Kode datetime_format yang umum tercantum di bawah ini. Anda juga dapat menggunakan kode datetime_format yang didukung oleh Python, datetime.strptime(). Pengimbangan zona waktu (%z) juga didukung meskipun itu tidak didukung sebelum python 3.2, [+-]HHMM tanpa titik dua(:). Untuk informasi selengkapnya, lihat Perilaku strftime() dan strptime().

%y: Tahun tanpa abad sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 00, 01,..., 99

%Y: Tahun dengan abad sebagai angka desimal.1970, 1988, 2001, 2013

%b: Bulan sebagai nama singkat lokal. Jan, Feb, ..., Des (id_ID);

%B: Bulan sebagai nama lengkap lokal. Januari, Februari, ..., Desember (id_ID);

%m: Bulan sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 01, 02,..., 12

%d: Tanggal dalam bulan sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 01, 02,..., 31

%H: Jam (24 jam) sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 00, 01,..., 23

%I: Jam (12 jam) sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 01, 02,..., 12

%p: Istilah lokal yang setara dengan AM atau PM.

%M: Menit sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 00, 01,..., 59

%S: Detik sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 00, 01,..., 59

%f: Mikrosekon sebagai angka desimal yang ditambah angka nol di depan angka satu digit (zero-padded). 000000,..., 999999

%z: Pengimbangan UTC dalam bentuk +HHMM atau -HHMM. +0000, -0400, +1030

Contoh format:

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

Menentukan zona stempel waktu log acara. Dua nilai yang didukung adalah UTC dan LOCAL. Default-nya adalah LOCAL, yang digunakan jika zona waktu tidak dapat disimpulkan berdasarkan datetime_format.

berkas

Menentukan file log yang ingin Anda push ke CloudWatch Log. File dapat menunjuk ke file tertentu atau beberapa file (menggunakan wildcard, seperti /var/log/system.log*). Hanya file terbaru yang didorong ke CloudWatch Log berdasarkan waktu modifikasi file. Kami sarankan Anda menggunakan wildcard untuk menentukan serangkaian file dengan jenis yang sama, seperti access_log.2014-06-01-01, access_log.2014-06-01-02, dan seterusnya, tetapi bukan beberapa jenis file, seperti access_log_80 dan access_log_443. Untuk menentukan beberapa jenis file, tambahkan entri pengaliran log lain ke file konfigurasi agar setiap jenis berkas log pergi ke pengaliran log yang berbeda. File terkompresi tidak didukung.

file_fingerprint_lines

Menentukan rentang baris untuk mengidentifikasi file. Nilai yang valid adalah satu angka atau dua angka yang dibatasi dengan tanda hubung, seperti '1', '2-5'. Nilai default-nya adalah '1' sehingga baris pertama digunakan untuk menghitung sidik jari. Garis sidik jari tidak dikirim ke CloudWatch Log kecuali semua baris yang ditentukan tersedia.

multi_line_start_pattern

Menentukan pola untuk mengidentifikasi awal pesan log. Pesan log dibuat dari baris yang sesuai dengan pola dan baris berikutnya yang tidak cocok dengan pola. Nilai yang valid adalah ekspresi reguler atau {datetime_format}. Jika menggunakan {datetime_format}, pilihan datetime_format harus ditentukan. Nilai default-nya adalah '^[^\ s]' sehingga semua baris yang dimulai dengan karakter yang bukan merupakan spasi kosong akan menutup pesan log sebelumnya dan memulai pesan log baru.

initial_position

Menentukan tempat untuk memulai membaca data (start_of_file atau end_of_file). Default-nya adalah start_of_file. Ini hanya digunakan jika tidak ada keadaan yang dipertahankan untuk pengaliran log tersebut.

encoding

Menentukan pengodean berkas log agar file dapat dibaca dengan benar. Default-nya adalah utf_8. Pengodean yang didukung oleh Python codecs.decode() dapat digunakan di sini.

> **Warning**
>
> Jika Anda menentukan pengodean yang salah, mungkin akan ada kehilangan data karena karakter yang tidak dapat didekode diganti dengan karakter lain.

Berikut adalah beberapa pengodean umum:

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737,
cp775, cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862,
cp863, cp864, cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950,
cp1006, cp1026, cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255,
cp1256, cp1257, cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr,
gb2312, gbk, gb18030, hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2,
iso2022_jp_2004, iso2022_jp_3, iso2022_jp_ext, iso2022_kr, latin_1,
iso8859_2, iso8859_3, iso8859_4, iso8859_5, iso8859_6, iso8859_7,
iso8859_8, iso8859_9, iso8859_10, iso8859_13, iso8859_14, iso8859_15,
iso8859_16, johab, koi8_r, koi8_u, mac_cyrillic, mac_greek, mac_iceland,
mac_latin2, mac_roman, mac_turkish, ptcp154, shift_jis, shift_jis_2004,
shift_jisx0213, utf_32, utf_32_be, utf_32_le, utf_16, utf_16_be,
utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

Menentukan durasi waktu untuk pembuatan batch log acara. Nilai minimumnya adalah 5000ms dan nilai default-nya adalah 5000ms.

batch_count

Menentukan jumlah maks log acara dalam batch, maksimum 10000. Nilai default-nya adalah 10000.

batch_size

Menentukan ukuran maks log acara dalam batch, dalam byte, maksimal 1048576 byte. Nilai default-nya adalah 1048576. Ukuran ini dihitung sebagai jumlah semua pesan kejadian dalam UTF-8, ditambah 26 byte untuk setiap log acara.

Menggunakan agen CloudWatch Log dengan proxy HTTP

Anda dapat menggunakan agen CloudWatch Log dengan proxy HTTP.

> **Note**
>
> Proksi HTTP didukung awslogs-agent-setup dalam.py versi 1.3.8 atau yang lebih baru.

Untuk menggunakan agen CloudWatch Log dengan proxy HTTP

1. Lakukan salah satu hal berikut ini:

   a. Untuk instalasi baru agen CloudWatch Log, jalankan perintah berikut:

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
   ```

   ```
   sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
   ```

   Untuk mempertahankan akses ke layanan metadata Amazon EC2 di instans EC2, gunakan --no-proxy 169.254.169.254 (disarankan). Untuk informasi selengkapnya, lihat Metadata Instans dan Data Pengguna di Panduan Pengguna Amazon EC2.

   Dalam nilai untuk `http-proxy` dan `https-proxy`, Anda menentukan seluruh URL.

   b. Untuk instalasi agen CloudWatch Log yang sudah ada, edit /var/awslogs/etc/proxy.conf, dan tambahkan proxy Anda:

   ```
   HTTP_PROXY=
   HTTPS_PROXY=
   NO_PROXY=
   ```

2. Restart agen agar perubahan diterapkan:

   ```
   sudo service awslogs restart
   ```

   Jika Anda menggunakan Amazon Linux 2, gunakan perintah berikut untuk me-restart agen:

   ```
   sudo service awslogsd restart
   ```

Membagi file konfigurasi agen CloudWatch Log

Jika Anda menggunakan awslogs-agent-setup .py versi 1.3.8 atau yang lebih baru dengan awscli-cwlogs 1.3.3 atau yang lebih baru, Anda dapat mengimpor konfigurasi aliran yang berbeda untuk berbagai komponen secara independen satu sama lain dengan membuat file konfigurasi tambahan

di direktori /var/awslogs/etc/config/. Ketika agen CloudWatch Log dimulai, itu mencakup konfigurasi aliran apa pun dalam file konfigurasi tambahan ini. Properti konfigurasi di bagian [general] harus didefinisikan dalam file konfigurasi utama (/var/awslogs/etc/awslogs.conf) dan diabaikan dalam file konfigurasi tambahan yang ditemukan di /var/awslogs/etc/config/.

Jika Anda tidak memiliki /var/awslogs/etc/config/ karena Anda menginstal agen dengan rpm, Anda dapat menggunakan direktori /etc/awslogs/config/ sebagai gantinya.

Restart agen agar perubahan diterapkan:

```
sudo service awslogs restart
```

Jika Anda menggunakan Amazon Linux 2, gunakan perintah berikut untuk me-restart agen:

```
sudo service awslogsd restart
```

CloudWatch FAQ agen log

Apa jenis rotasi file yang didukung?

Mekanisme rotasi file berikut didukung:

- Mengganti nama berkas log yang ada dengan akhiran numerik, kemudian membuat ulang berkas log kosong asli. Misalnya, /var/log/syslog.log diganti namanya menjadi /var/log/syslog.log.1. Jika /var/log/syslog.log.1 sudah ada dari rotasi sebelumnya, namanya diganti menjadi /var/log/syslog.log.2.
- Memotong berkas log asli di tempat setelah membuat salinan. Misalnya, /var/log/syslog.log disalin ke /var/log/syslog.log.1 dan /var/log/syslog.log dipotong. Mungkin akan ada kehilangan data untuk kasus ini, jadi berhati-hatilah dalam menggunakan mekanisme rotasi file ini.
- Membuat file baru dengan pola umum seperti yang lama. Misalnya, /var/log/syslog.log.2014-01-01 tetap ada dan /var/log/syslog.log.2014-01-02 dibuat.

Sidik jari (ID sumber) file dihitung dengan hashing kunci pengaliran log dan baris pertama dari konten file. Untuk menggantikan perilaku ini, pilihan file_fingerprint_lines dapat digunakan. Ketika rotasi file terjadi, file baru seharusnya memiliki konten baru dan file lama tidak seharusnya memiliki tambahan konten; agen mendorong file baru setelah selesai membaca file lama.

Bagaimana cara menentukan versi agen yang saya gunakan?

Jika Anda menggunakan skrip penyiapan untuk menginstal agen CloudWatch Logs, Anda dapat menggunakan /var/awslogs/bin/awslogs-version.sh untuk memeriksa versi agen yang Anda gunakan. Versi agen dan dependensi utamanya akan dicetak. Jika Anda menggunakan yum untuk menginstal agen CloudWatch Log, Anda dapat menggunakan “yum info awslogs” dan “yum info aws-cli-plugin-cloudwatch -logs” untuk memeriksa versi agen dan plugin Logs. CloudWatch

Bagaimana entri log dikonversi menjadi log acara?

Log acara berisi dua properti: stempel waktu ketika peristiwa terjadi, dan pesan log mentah. Secara default, semua baris yang dimulai dengan karakter yang bukan spasi kosong akan menutup pesan log sebelumnya, jika ada, dan memulai pesan log baru. Untuk menggantikan perilaku ini, multi_line_start_pattern dapat digunakan dan setiap baris yang cocok dengan pola akan memulai pesan log baru. Pola bisa berupa regex atau '{datetime_format}'. Sebagai contoh, jika baris pertama dari setiap pesan log berisi stempel waktu, seperti '2014-01-02T13:13:01Z', multi_line_start_pattern dapat diatur ke '\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z'. Untuk menyederhanakan konfigurasi, variabel '{datetime_format}' dapat digunakan jika datetime_format option ditentukan. Untuk contoh yang sama, jika datetime_format diatur ke '%Y-%m-%dT%H:%M:%S%z', multi_line_start_pattern dapat berupa '{datetime_format}'.

Waktu saat ini akan digunakan untuk setiap log acara jika datetime_format tidak disediakan. Jika datetime_format yang diberikan tidak valid untuk pesan log tertentu, stempel waktu dari log acara terakhir dengan stempel waktu yang berhasil diurai akan digunakan. Jika tidak ada log acara sebelumnya, waktu saat ini akan digunakan. Pesan peringatan akan dicatat ketika peristiwa log kembali ke waktu saat ini atau waktu log acara sebelumnya.

Stempel waktu digunakan untuk mengambil log acara dan menghasilkan metrik, jadi jika Anda menentukan format yang salah, log acara mungkin tidak bisa diambil dan akan menghasilkan metrik yang salah.

Bagaimana batch log acara dibuat?

Suatu batch akan menjadi penuh dan dipublikasikan ketika salah satu dari persyaratan berikut terpenuhi:

1. Parameter jumlah waktu buffer_duration telah berlalu sejak log acara pertama ditambahkan.
2. Kurang dari batch_size log acara telah terakumulasi, tetapi menambahkan log acara baru akan melampaui batch_size.
3. Jumlah log acara telah mencapai batch_count.

4. Log acara dari batch tidak berlangsung lebih dari 24 jam, tetapi menambahkan log acara baru akan melampaui batas 24 jam.

Apa yang menyebabkan entri log, log acara, atau batch dilewati atau dipotong?

Untuk mematuhi batasan operasi `PutLogEvents`, masalah berikut dapat menyebabkan log acara atau batch dilewati.

> **Note**
>
> Agen CloudWatch Logs menulis peringatan ke lognya saat data dilewati.

1. Jika ukuran log acara melebihi 256 KB, log acara akan dilewati sepenuhnya.
2. Jika stempel waktu log acara menyatakan waktu yang lebih dari 2 jam mendatang, log acara akan dilewati.
3. Jika stempel waktu log acara menyatakan waktu yang lebih dari 14 hari yang lampau, log acara akan dilewati.
4. Jika log acara lebih tua dari periode retensi grup log, seluruh batch akan dilewati.
5. Jika batch log acara dalam satu permintaan `PutLogEvents` mencakup lebih dari 24 jam, operasi `PutLogEvents` akan gagal.

Apakah menghentikan agen akan menyebabkan kehilangan data/duplikat?

Tidak, selama file state tersedia dan tidak ada rotasi file yang terjadi sejak terakhir dijalankan. Agen CloudWatch Log dapat memulai dari tempat berhenti dan terus mendorong data log.

Dapatkah saya mengarahkan berkas log yang berbeda dari host yang sama atau berbeda ke pengaliran log yang sama?

Mengonfigurasi beberapa sumber log untuk mengirim data ke satu pengaliran log tidaklah didukung.

Panggilan API apa yang dilakukan agen (atau tindakan apa yang harus saya tambahkan ke kebijakan IAM saya)?

Agen CloudWatch Log membutuhkan`CreateLogGroup`,`CreateLogStream`,`DescribeLogStreams`, dan `PutLogEvents` operasi. Jika Anda menggunakan agen terbaru, `DescribeLogStreams` tidak diperlukan. Lihat contoh kebijakan IAM di bawah ini.

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Saya tidak ingin agen CloudWatch Log membuat grup log atau aliran log secara otomatis. Bagaimana cara mencegah agen membuat grup log dan pengaliran log?

Dalam kebijakan IAM, Anda dapat membatasi agen hanya ke operasi berikut: `DescribeLogStreams`, `PutLogEvents`.

Sebelum Anda mencabut izin `CreateLogGroup` dan `CreateLogStream` dari agen, pastikan untuk membuat grup log dan pengaliran log yang Anda inginkan untuk digunakan oleh agen. Agen log tidak dapat membuat pengaliran log dalam grup log yang telah Anda buat kecuali memiliki izin `CreateLogGroup` dan `CreateLogStream`.

Log apa yang harus saya lihat saat memecahkan masalah?

Log penginstalan agen berada di `/var/log/awslogs-agent-setup.log` dan log agen berada di `/var/log/awslogs.log`.

Pemantauan dengan CloudWatch metrik

CloudWatch Log mengirimkan metrik ke Amazon CloudWatch setiap menit.

CloudWatch Metrik log

Namespace `AWS/Logs` mencakup metrik berikut.

Metrik	Deskripsi
`CallCount`	Jumlah operasi API tertentu yang dilakukan di akun Anda. `CallCount` adalah metrik penggunaan layanan CloudWatch Log. Untuk informasi selengkapnya, lihat CloudWatch Metrik penggunaan layanan log. Dimensi yang Valid: Kelas, Sumber Daya, Layanan, Jenis Statistik Valid: Sum Unit: Tidak ada
`DeliveryErrors`	Jumlah peristiwa log yang CloudWatch Log menerima kesalahan saat meneruskan data ke tujuan langganan. Jika layanan tujuan mengembalikan kesalahan yang dapat dicoba ulang seperti pengecualian pembatasan atau pengecualian layanan yang dapat dicoba ulang (misalnya HTTP 5xx), CloudWatch Log terus mencoba lagi pengiriman hingga 24 jam. CloudWatch Log tidak mencoba mengirim ulang jika kesalahan adalah kesalahan yang tidak dapat dicoba ulang, seperti atau. `AccessDeniedException` `ResourceNotFoundException` Dimensi yang Valid: LogGroupName DestinationType,, FilterName, PolicyLevel Statistik Valid: Sum Unit: Tidak ada

Metrik	Deskripsi
`DeliveryThrottling`	Jumlah peristiwa log yang CloudWatch Log dibatasi saat meneruskan data ke tujuan langganan. Jika layanan tujuan mengembalikan kesalahan yang dapat dicoba ulang seperti pengecualian pembatasan atau pengecualian layanan yang dapat dicoba ulang (misalnya HTTP 5xx), CloudWatch Log terus mencoba lagi pengiriman hingga 24 jam. CloudWatch Log tidak mencoba mengirim ulang jika kesalahan adalah kesalahan yang tidak dapat dicoba ulang, seperti atau. `AccessDeniedException` `ResourceNotFoundException` Dimensi yang Valid: LogGroupName DestinationType,, FilterName, PolicyLevel Statistik Valid: Sum Unit: Tidak ada
`EMFParsingErrors`	Jumlah kesalahan penguraian yang ditemui saat memproses log format metrik yang disematkan. Kesalahan seperti itu terjadi ketika log diidentifikasi sebagai format metrik tertanam tetapi tidak mengikuti format yang benar. Untuk informasi selengkapnya tentang format metrik yang disematkan, lihat Spesifikasi: Format metrik tertanam. Dimensi yang Benar: `LogGroupName` Statistik Valid: Sum Unit: Tidak ada

Metrik	Deskripsi
`EMFValidationErrors`	Jumlah kesalahan validasi yang ditemui saat memproses log format metrik tertanam. Kesalahan ini terjadi ketika definisi metrik dalam log format metrik yang disematkan tidak mematuhi format dan `MetricDatum` spesifikasi metrik yang disematkan. Untuk informasi tentang format metrik yang CloudWatch disematkan, lihat Spesifikasi: Format metrik tertanam. Untuk informasi tentang tipe data`MetricDatum` , lihat MetricDatumdi Referensi Amazon CloudWatch API. **Note** Kesalahan validasi tertentu dapat menyebabkan beberapa metrik dalam log EMF tidak dipublikasikan. Misalnya, semua metrik yang disetel dengan namespace yang tidak valid akan dihapus. Dimensi yang Benar: `LogGroupName` Statistik Valid: Sum Unit: Tidak ada
`ErrorCount`	Jumlah operasi API yang dilakukan di akun Anda yang mengakibatkan kesalahan. `ErrorCount` adalah metrik penggunaan layanan CloudWatch Log. Untuk informasi selengkapnya, lihat CloudWatch Metrik penggunaan layanan log. Dimensi yang Valid: Kelas, Sumber Daya, Layanan, Jenis Statistik Valid: Sum Unit: Tidak ada

Metrik	Deskripsi
`ForwardedBytes`	Volume log acara dalam byte terkompresi yang diteruskan ke tujuan langganan. Dimensi yang Valid: LogGroupName, DestinationType, FilterName Statistik Valid: Sum Unit: Byte
`Forwarded LogEvents`	Jumlah log acara yang diteruskan ke tujuan langganan. Dimensi yang Valid: LogGroupName DestinationType,, FilterName, PolicyLevel Statistik Valid: Sum Unit: Tidak ada
`IncomingBytes`	Volume peristiwa log dalam byte tidak terkompresi yang diunggah ke Log. CloudWatch Ketika digunakan dengan dimensi `LogGroupName` , ini adalah volume log acara dalam byte tak terkompresi yang diunggah ke grup log. Dimensi yang Valid: LogGroupName Statistik Valid: Sum Unit: Bita
`IncomingL ogEvents`	Jumlah peristiwa log yang diunggah ke CloudWatch Log. Ketika digunakan dengan dimensi `LogGroupName` , ini adalah jumlah log acara yang diunggah ke grup log. Dimensi yang Valid: LogGroupName Statistik Valid: Sum Unit: Tidak ada

Metrik	Deskripsi
`LogEvents WithFindings`	Jumlah peristiwa log yang cocok dengan string data yang Anda audit menggunakan fitur perlindungan data CloudWatch Log. Untuk informasi selengkapnya, lihat Membantu melindungi data log sensitif dengan masking. Dimensi valid: Tidak Ada Statistik Valid: Sum Unit: Tidak ada
`ThrottleCount`	Jumlah operasi API yang dilakukan di akun Anda yang dibatasi karena kuota penggunaan. `ThrottleCount` adalah metrik penggunaan layanan CloudWatch Log. Untuk informasi selengkapnya, lihat CloudWatch Metrik penggunaan layanan log. Dimensi yang Valid: Kelas, Sumber Daya, Layanan, Jenis Statistik Valid: Sum Unit: Tidak ada

Dimensi untuk metrik CloudWatch Log

Dimensi yang dapat Anda gunakan dengan metrik CloudWatch Log tercantum dalam tabel berikut.

Dimensi	Deskripsi
`LogGroupName`	Nama grup CloudWatch log Log untuk menampilkan metrik.
`DestinationType`	Tujuan berlangganan untuk data CloudWatch Log, yang dapat berupa AWS Lambda, Amazon Kinesis Data Streams, atau Amazon Data Firehose.

Dimensi	Deskripsi
`FilterName`	Nama filter langganan yang meneruskan data dari grup log ke tujuan. Nama filter langganan secara otomatis dikonversi CloudWatch menjadi ASCII dan karakter apa pun yang tidak didukung diganti dengan tanda tanya (?).

Dimensi untuk metrik yang terkait dengan filter langganan tingkat akun tercantum dalam tabel berikut.

Dimensi	Deskripsi
`PolicyLevel`	Tingkat di mana kebijakan berlaku. Saat ini, satu-satunya nilai yang valid untuk dimensi ini adalah `AccountPolicy`
`DestinationType`	Tujuan berlangganan untuk data CloudWatch Log, yang dapat berupa AWS Lambda, Amazon Kinesis Data Streams, atau Amazon Data Firehose.
`FilterName`	Nama filter langganan yang meneruskan data dari grup log ke tujuan. Nama filter langganan secara otomatis dikonversi CloudWatch menjadi ASCII dan karakter apa pun yang tidak didukung diganti dengan tanda tanya (?).

CloudWatch Metrik penggunaan layanan log

CloudWatch Log mengirimkan metrik untuk CloudWatch melacak operasi API CloudWatch Log penggunaan. Metrik ini sesuai dengan kuota AWS layanan. Dengan melacak metrik-metrik tersebut dapat membantu Anda mengelola kuota secara proaktif. Untuk informasi selengkapnya, lihat Integrasi Service Quotas dan Metrik Penggunaan.

Misalnya, Anda dapat melacak `ThrottleCount` metrik atau mengatur alarm pada metrik itu. Jika nilai metrik ini naik, Anda harus mempertimbangkan untuk meminta peningkatan kuota untuk operasi API yang terhambat. Untuk informasi selengkapnya tentang kuota layanan CloudWatch Log, lihatCloudWatch Kuota log.

CloudWatch Log menerbitkan metrik penggunaan kuota layanan setiap menit di ruang nama dan ruang nama. `AWS/Usage` `AWS/Logs`

Tabel berikut mencantumkan metrik penggunaan layanan yang diterbitkan oleh CloudWatch Log. Metrik ini tidak memiliki unit tertentu. Statistik yang paling berguna untuk metrik ini adalah`SUM`, yang mewakili jumlah operasi total untuk periode 1 menit.

Masing-masing metrik ini diterbitkan dengan nilai untuk semua`Service`,, `ClassType`, dan `Resource` dimensi. Mereka juga diterbitkan dengan satu dimensi yang disebut`Account Metrics`. Gunakan `Account Metrics` dimensi untuk melihat jumlah metrik untuk semua operasi API di akun Anda. Gunakan dimensi lain dan tentukan nama operasi API untuk `Resource` dimensi tersebut guna menemukan metrik untuk API tertentu.

Metrik-metrik

Metrik	Deskripsi
`CallCount`	Jumlah operasi tertentu yang dilakukan di akun Anda. `CallCount` diterbitkan di `AWS/Logs` ruang nama `AWS/Usage` dan ruang nama.
`ErrorCount`	Jumlah operasi API yang dilakukan di akun Anda yang mengakibatkan kesalahan. `ErrorCount` diterbitkan hanya dalam`AWS/Logs`.
`ThrottleCount`	Jumlah operasi API yang dilakukan di akun Anda yang dibatasi karena kuota penggunaan. `ThrottleCount` diterbitkan hanya dalam`AWS/Logs`.

Dimensi

Dimensi	Deskripsi
`Account metrics`	Gunakan dimensi ini untuk mendapatkan jumlah metrik di semua API CloudWatch Log. Jika Anda ingin melihat metrik untuk satu API tertentu, gunakan dimensi lain yang tercantum dalam tabel ini dan tentukan nama API sebagai nilai`Resource`.

Dimensi	Deskripsi
`Service`	Nama AWS layanan yang berisi sumber daya. Untuk metrik penggunaan CloudWatch Log, nilai untuk dimensi ini adalah`Logs`.
`Class`	Kelas sumber daya yang dilacak. CloudWatch Metrik penggunaan API log menggunakan dimensi ini dengan nilai. `None`
`Type`	Jenis sumber daya yang sedang ditelusuri. Saat ini, ketika dimensi `Service` adalah `Logs`, satu-satunya nilai yang benar untuk `Type` adalah `API`.
`Resource`	Nama operasi API. Nilai yang valid mencakup semua nama operasi API yang tercantum dalam Tindakan. Misalnya, `PutLogEvents`

CloudWatch Kuota log

Tabel berikut menyediakan kuota layanan default, juga disebut sebagai batas, untuk CloudWatch Log untuk AWS akun. Sebagian besar kuota layanan ini, tetapi tidak semua, terdaftar di bawah namespace Amazon CloudWatch Logs di konsol Service Quotas. Untuk meminta peningkatan kuota tersebut, lihat prosedurnya nanti di bagian ini.

Sumber Daya	Kuota bawaan
Kebijakan tingkat akun	Kebijakan filter langganan satu tingkat akun per akun. Satu kebijakan perlindungan data tingkat akun per akun. Kuota-kuota ini tidak dapat diubah.
Detektor anomali	10 detektor anomali per akun. Kuota ini tidak dapat diubah.
Ukuran batch	Ukuran batch maksimum adalah 1.048.576 byte. Ukuran ini dihitung sebagai jumlah semua pesan kejadian dalam UTF-8, ditambah 26 byte untuk setiap log acara. Kuota ini tidak dapat diubah.
Pengarsipan data	Pengarsipan data hingga 5 GB secara gratis. Kuota ini tidak dapat diubah.
CreateLogGroup	10 transaksi per detik (TPS/akun/wilayah), setelah itu transaksi dibatasi. Anda dapat meminta penambahan kuota.
CreateLogStream	50 transaksi per detik (TPS/akun/Wilayah), setelahnya transaksi tersebut akan mengalami throttling. Anda dapat meminta penambahan kuota.
Pengidentifikasi data khusus	Setiap kebijakan perlindungan data dapat mencakup hingga 10 pengidentifikasi data kustom. Anda dapat meminta penambahan kuota.

Sumber Daya	Kuota bawaan
	Setiap ekspresi reguler yang mendefinisikan pengenal data kustom dapat menyertakan hingga 200 karakter. Kuota ini tidak dapat diubah.
DeleteLogGroup	10 transaksi per detik (TPS/akun/wilayah), setelah itu transaksi dibatasi. Anda dapat meminta penambahan kuota.
DeleteLogStream	15 transaksi per detik (TPS/akun/wilayah), setelah itu transaksi dibatasi. Anda dapat meminta penambahan kuota.
DescribeLogGroups	10 transaksi per detik (TPS/Akun/wilayah). Anda dapat meminta penambahan kuota.
DescribeLogStreams	25 transaksi per detik (TPS/Akun/wilayah). Anda dapat meminta penambahan kuota.
Bidang log yang ditemukan	CloudWatch Wawasan Log dapat menemukan maksimal 1000 bidang peristiwa log dalam grup log. Kuota ini tidak dapat diubah. Untuk informasi selengkapnya, lihat Log yang didukung dan bidang yang ditemukan.
Bidang log yang diekstraksi dalam log JSON	CloudWatch Wawasan Log dapat mengekstrak maksimal 200 bidang peristiwa log dari log JSON. Kuota ini tidak dapat diubah. Untuk informasi selengkapnya, lihat Log yang didukung dan bidang yang ditemukan.
Tugas ekspor	Satu tugas ekspor aktif (berjalan atau tertunda) pada satu waktu, per akun. Kuota ini tidak dapat diubah.

Sumber Daya	Kuota bawaan
FilterLogEvents	25 permintaan per detik di AS Timur (Virginia N.) 5 permintaan per detik di Wilayah berikut: • Asia Pasifik (Jakarta) • Asia Pasifik (Osaka) • Eropa (Frankfurt) • Kanada Barat (Calgary) • Israel (Tel Aviv) 10 permintaan per detik di Wilayah lain. Kuota ini tidak dapat diubah.

Sumber Daya	Kuota bawaan
GetLogEvents	30 permintaan per detik di Eropa (Paris). 10 permintaan per detik di Wilayah berikut: • AS Barat (Oregon) • Asia Pasifik (Jakarta) • Asia Pasifik (Osaka) • Kanada Barat (Calgary) • Eropa (Irlandia) • Eropa (Frankfurt) • Israel (Tel Aviv) 25 permintaan per detik di semua Wilayah lainnya. Kuota ini tidak dapat diubah. Kami merekomendasikan langganan jika Anda terus memproses data baru. Jika Anda membutuhkan data historis, sebaiknya Anda mengekspor data ke Amazon S3.
Data masuk	Hingga 5 GB data masuk secara gratis. Kuota ini tidak dapat diubah.
Sesi bersamaan Live Tail.	15 sesi bersamaan. Anda dapat meminta penambahan kuota.
Live Tail: grup log dicari dalam satu sesi.	Maksimal 10 grup log yang dipindai dalam satu sesi Live Tail. Kuota ini tidak dapat diubah.
Ukuran acara log	256 KB (maksimum). Kuota ini tidak dapat diubah.

Sumber Daya	Kuota bawaan
Grup log	1.000.000 grup log per akun per Wilayah. Anda dapat meminta kenaikan kuota. Tidak ada kuota pada jumlah pengaliran log yang dapat menjadi milik satu grup log.
Filter metrik	100 per grup log. Kuota ini tidak dapat diubah.
Metrik format metrik tertanam	100 metrik per peristiwa log dan 30 dimensi per metrik. Untuk informasi selengkapnya tentang format metrik yang disematkan, lihat Spesifikasi: Format Metrik Tertanam di Panduan CloudWatch Pengguna Amazon.
PutLogEvents	Ukuran batch maksimum PutLogEvents permintaan adalah 1MB. Ukuran ini dihitung sebagai jumlah semua pesan kejadian dalam UTF-8, ditambah 26 byte untuk setiap log acara. 5000 transaksi per detik per akun per Wilayah Anda dapat meminta kenaikan kuota throttling per detik dengan menggunakan layanan. Service Quotas
Batas waktu eksekusi kueri	Waktu kueri di CloudWatch Logs Insights habis setelah 60 menit. Batas waktu ini tidak dapat diubah.
Grup log yang dikueri	Maksimal 50 grup log dapat ditanyakan dalam satu kueri Wawasan CloudWatch Log. Kuota ini tidak dapat diubah.
Konkurensi kueri	Untuk grup log kelas Standar, maksimal 30 kueri Wawasan CloudWatch Log bersamaan, termasuk kueri yang telah ditambahkan ke dasbor. Untuk grup log kelas Akses Jarang, maksimal 5 kueri Wawasan CloudWatch Log bersamaan, termasuk kueri yang telah ditambahkan ke dasbor. Kuota-kuota ini tidak dapat diubah.

Sumber Daya	Kuota bawaan
Kueri yang dihasilkan dari bahasa alami	Sebanyak lima permintaan kueri yang dihasilkan bahasa alami bersamaan.
Ketersediaan kueri	Kueri yang dibangun di konsol tersedia selama 30 hari, melalui perintah History. Periode ketersediaan ini tidak dapat diubah. Definisi kueri yang dibuat dengan menggunakan PutQueryDefinitiontidak kedaluwarsa.
Ketersediaan hasil kueri	Hasil dari kueri dapat diperoleh selama 7 hari. Waktu ketersediaan ini tidak dapat diubah.
Hasil kueri ditampilkan di konsol	Secara default, hingga 1000 baris hasil kueri ditampilkan di konsol. Anda dapat menggunakan perintah **`limit`** dalam kueri untuk meningkatkan ini hingga sebanyak 10.000 baris. Untuk informasi selengkapnya, lihat CloudWatch Sintaks kueri Log Insights.
Ekspresi reguler	Hingga 5 pola filter yang berisi ekspresi reguler untuk setiap grup log saat membuat filter metrik atau filter langganan. Kuota ini tidak dapat diubah. Hingga 2 ekspresi reguler untuk setiap pola filter, saat membuat pola filter terbatas atau JSON untuk filter metrik dan filter langganan atau saat memfilter peristiwa log.
Kebijakan sumber daya	Hingga 10 kebijakan sumber daya CloudWatch Log per Wilayah per akun. Kuota ini tidak dapat diubah.
Kueri tersimpan	Anda dapat menyimpan sebanyak 1000 kueri Wawasan CloudWatch Log, per Wilayah per akun. Kuota ini tidak dapat diubah.
Filter langganan	2 per grup log. Kuota ini tidak dapat diubah.

Mengelola kuota layanan CloudWatch Log

CloudWatch Log telah terintegrasi dengan Service Quotas, sebuah AWS layanan yang memungkinkan Anda untuk melihat dan mengelola kuota Anda dari lokasi pusat. Untuk informasi selengkapnya, lihat Apa itu Service Quotas? di Panduan Pengguna Service Quotas.

Service Quotas memudahkan untuk mencari nilai kuota layanan CloudWatch Log Anda.

AWS Management Console

Untuk melihat kuota layanan CloudWatch Log menggunakan konsol

1. Buka konsol Service Quotas di https://console.aws.amazon.com/servicequotas/.
2. Di panel navigasi, pilih Layanan AWS .
3. Dari daftar AWS layanan, cari dan pilih Amazon CloudWatch Logs.

   Dalam daftar service quotas, Anda dapat melihat nama service quotas, nilai terapan (jika tersedia), kuota default AWS , dan apakah nilai kuota dapat disesuaikan.
4. Untuk melihat informasi tambahan tentang service quotas, seperti deskripsi, pilih nama kuota.
5. (Opsional) Untuk meminta peningkatan kuota, pilih kuota yang ingin Anda tingkatkan, pilih Request quota increase (Meminta kenaikan kuota), masukkan atau pilih informasi yang diperlukan, dan pilih Request (Permintaan).

Untuk bekerja lebih lanjut dengan kuota layanan menggunakan konsol lihat Panduan Pengguna Service Quotas. Untuk meminta kenaikan kuota, lihat Meminta kenaikan kuota dalam Panduan Pengguna Service Quotas.

AWS CLI

Untuk melihat kuota layanan CloudWatch Log menggunakan AWS CLI

Jalankan perintah berikut untuk melihat kuota CloudWatch Log default.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code logs \
    --output table
```

Untuk bekerja lebih banyak dengan kuota layanan menggunakan AWS CLI, lihat Referensi Perintah Service AWS CLI Quotas. Untuk meminta kenaikan kuota, lihat perintah `request-service-quota-increase` di Referensi Perintah AWS CLI.

Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
CloudWatch Dukungan Wawasan Log untuk pembuatan kueri bahasa alami umumnya tersedia	CloudWatch Logs Insights mendukung bahasa alami untuk menghasilkan dan memperbarui kueri. Untuk informasi selengkapnya, lihat Menggunakan bahasa alami untuk membuat dan memperbarui kueri Wawasan CloudWatch Log.	Juni 20, 2024
CloudWatchLogsRead OnlyAccesskebijakan diperbarui	CloudWatch Log menambahkan `cloudwatch:GenerateQuery` izin ke CloudWatchLogsReadOnlyAccess, sehingga pengguna dengan kebijakan ini dapat menghasilkan string kueri Wawasan CloudWatch Log dari prompt bahasa alami.	26 November 2023
CloudWatchLogsFull Accesskebijakan diperbarui	CloudWatch Log menambahkan `cloudwatch:GenerateQuery` izin ke CloudWatchLogsFullAccess, sehingga pengguna dengan kebijakan ini dapat menghasilkan string kueri Wawasan CloudWatch Log dari prompt bahasa alami.	26 November 2023

CloudWatch Log menambahkan analisis pola log	CloudWatch Log sekarang memindai pola dalam peristiwa log setiap kali Anda melakukan kueri Wawasan CloudWatch Log. Untuk informasi lebih lanjut, lihat Analisis pola.	26 November 2023
CloudWatch Log menambahkan deteksi anomali log	Anda dapat membuat detektor anomali log untuk grup log. Detektor anomali memindai peristiwa log yang dicerna ke dalam grup log dan menemukan anomali dalam data log. Untuk informasi lebih lanjut, lihat Deteksi anomali log.	26 November 2023
CloudWatch Log menambahkan fitur bandingkan	Anda sekarang dapat menggunakan Wawasan CloudWatch Log untuk membandingkan perubahan dalam peristiwa log Anda dari waktu ke waktu. Untuk informasi selengkapnya, lihat Bandingkan (diff) dengan rentang waktu sebelumnya.	26 November 2023

CloudWatch Log menambahkan kelas log baru	CloudWatch Log mendukung dua kelas grup log sehingga Anda dapat memiliki opsi hemat biaya untuk log yang jarang Anda akses, dan Anda juga memiliki opsi fitur lengkap untuk log yang memerlukan pemantauan waktu nyata atau fitur lainnya. Untuk informasi selengkapnya, lihat Kelas log.	26 November 2023
CloudWatch Logs Insights mendukung pembuatan kueri bahasa alami	CloudWatch Logs Insights mendukung bahasa alami untuk menghasilkan dan memperbarui kueri. Untuk informasi selengkapnya, lihat Menggunakan bahasa alami untuk membuat dan memperbarui kueri Wawasan CloudWatch Log.	26 November 2023
CloudWatch Log menambahkan dukungan sintaks pola filter ekspresi reguler untuk Live Tail	Sekarang Anda dapat menyesuaikan operasi pencarian dan pencocokan lebih lanjut untuk memenuhi kebutuhan Anda dengan ekspresi reguler yang fleksibel dalam pola filter Live Tail. Untuk informasi selengkapnya, lihat Memfilter sintaks pola di Panduan Pengguna CloudWatch Log Amazon.	13 November 2023

CloudWatch Log menambahkan dukungan sintaks pola filter ekspresi reguler untuk filter metrik, filter langganan, dan peristiwa log filter	Anda sekarang dapat menyesuaikan operasi pencarian dan pencocokan lebih lanjut untuk memenuhi kebutuhan Anda dengan ekspresi reguler yang fleksibel dalam pola filter. Untuk informasi selengkapnya, lihat Memfilter sintaks pola di Panduan Pengguna CloudWatch Log Amazon.	5 September 2023
CloudWatch Log Insights menambahkan perintah pola	Sekarang Anda dapat menggunakan pola dalam kueri Wawasan CloudWatch Log untuk secara otomatis mengelompokkan data log Anda ke dalam pola. Pola adalah struktur teks bersama yang berulang di antara bidang log Anda. Untuk informasi selengkapnya, lihat pola di Panduan Pengguna CloudWatch Log Amazon.	Juli 17, 2023
CloudWatch Logs Insights menambahkan perintah dedup	Sekarang Anda dapat menggunakan dedup dalam kueri Wawasan CloudWatch Log untuk menghapus hasil duplikat berdasarkan nilai tertentu di bidang yang Anda tentukan. Untuk informasi selengkapnya, lihat dedup di Panduan Pengguna Amazon CloudWatch Logs.	20 Juni 2023

Kebijakan perlindungan data tingkat akun	Anda sekarang dapat menetapkan kebijakan perlindungan data di tingkat akun. Kebijakan tingkat akun ini dapat mengaudit dan menutupi informasi sensitif dalam peristiwa log di semua grup log di akun. Untuk informasi selengkapnya, lihat Membantu melindungi data log sensitif dengan masking di Panduan Pengguna Amazon CloudWatch Logs.	8 Juni 2023
Fitur Live Tail ditambahkan	CloudWatch Log menambahkan kemampuan Live Tail, sehingga Anda dapat memindai log saat tertelan untuk membantu pemecahan masalah. Anda dapat secara opsional memfilter aliran peristiwa log yang ditampilkan berdasarkan istilah yang ditentukan, dan juga menyorot peristiwa log yang memiliki istilah tertentu. Untuk informasi selengkapnya, silakan lihat Menggunakan live tail untuk melihat log mendekati waktu nyata.	6 Juni 2023

CloudWatchLogsReadOnlyAccesskebijakan diperbarui	CloudWatch Log menambahkan izin ke CloudWatchLogsReadOnlyAccess. Izin `logs:StartLiveTail` dan `logs:StopLiveTail` izin ditambahkan sehingga pengguna dengan kebijakan ini dapat menggunakan konsol untuk memulai dan menghentikan sesi ekor langsung CloudWatch Log. Untuk informasi selengkapnya, silakan lihat Menggunakan live tail untuk melihat log mendekati waktu nyata.	6 Juni 2023
CloudWatch Log Insights dirilis	Anda dapat menggunakan Wawasan CloudWatch Log untuk mencari dan menganalisis data log secara interaktif. Untuk informasi selengkapnya lihat Menganalisis Data CloudWatch Log dengan Wawasan Log di Panduan Pengguna CloudWatch Log Amazon	27 November 2018

Dukungan untuk titik akhir VPC Amazon VPC	Anda sekarang dapat membuat koneksi pribadi antara VPC dan CloudWatch Log Anda. Untuk informasi selengkapnya, lihat Menggunakan CloudWatch Log dengan Titik Akhir VPC Antarmuka di Panduan Pengguna Amazon CloudWatch Logs.	28 Juni 2018

Tabel berikut menjelaskan perubahan penting pada Panduan Pengguna Amazon CloudWatch Logs.

Perubahan	Deskripsi	Tanggal rilis
Titik akhir VPC antarmuka	Di beberapa Wilayah, Anda dapat menggunakan titik akhir VPC antarmuka untuk menjaga lalu lintas antara VPC Amazon dan Log Anda CloudWatch agar tidak meninggalkan jaringan Amazon. Untuk informasi selengkapnya, lihat Menggunakan CloudWatch Log dengan titik akhir VPC antarmuka.	Selasa, 07 Maret 2018
Log kueri DNS Route 53	Anda dapat menggunakan CloudWatch Log untuk menyimpan log tentang kueri DNS yang diterima oleh Route 53. Untuk informasi selengkapnya, lihat Apa itu Amazon CloudWatch Logs? atau Mencatat Log Kueri DNS dalam Panduan Developer Amazon Route 53.	7 September 2017
Menandai grup log	Anda dapat menggunakan tanda untuk mengategorikan grup log Anda. Untuk informasi selengkapnya, lihat Tandai grup log di Amazon CloudWatch Logs.	13 Desember 2016

Perubahan	Deskripsi	Tanggal rilis
Penyempurnaan konsol	Anda dapat menavigasi dari grafik metrik ke grup log terkait. Untuk informasi selengkapnya, lihat Pivot dari metrik ke log.	Selasa, 07 Nopember 2016
Penyempurnaan kegunaan konsol	Meningkatkan pengalaman agar lebih mudah mencari, memfilter, dan memecahkan masalah. Misalnya, Anda sekarang dapat memfilter data log Anda berdasarkan rentang tanggal dan waktu. Untuk informasi selengkapnya, lihat Lihat data log yang dikirim ke CloudWatch Log.	Selasa, 29 Agustus 2016
Menambahkan AWS CloudTrail dukungan untuk Amazon CloudWatch Log dan metrik CloudWatch Log baru	Ditambahkan AWS CloudTrail dukungan untuk CloudWatch Log. Untuk informasi selengkapnya, lihat Logging CloudWatch Logs API dan operasi konsol di AWS CloudTrail.	10 Maret 2016
Menambahkan dukungan untuk ekspor CloudWatch Log ke Amazon S3	Menambahkan dukungan untuk mengekspor data CloudWatch Log ke Amazon S3. Untuk informasi selengkapnya, lihat Mengekspor data log ke Amazon S3.	Selasa, 07 Desember 2015
Menambahkan dukungan untuk peristiwa yang AWS CloudTrail dicatat di Amazon CloudWatch Logs	Anda dapat membuat alarm CloudWatch dan menerima notifikasi aktivitas API tertentu seperti yang ditangkap oleh CloudTrail dan menggunakan notifikasi untuk melakukan pemecahan masalah.	10 November 2014

Perubahan	Deskripsi	Tanggal rilis
Ditambahkan dukungan untuk Amazon CloudWatch Logs	Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan mengakses sistem, aplikasi, dan file log kustom Anda dari instans Amazon Elastic Compute Cloud (Amazon EC2) atau sumber lain. Anda kemudian dapat mengambil data log terkait dari CloudWatch Log menggunakan CloudWatch konsol Amazon, perintah CloudWatch Log di AWS CLI, atau CloudWatch Logs SDK. Untuk informasi selengkapnya, lihat Apa itu Amazon CloudWatch Logs?.	10 Juli 2014

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS

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