



Panduan Pengguna

CloudWatch Log Amazon



CloudWatch Log Amazon: Panduan Pengguna

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau mungkin tidak.

Table of Contents

Apa itu Amazon CloudWatch Logs?	1
Fitur	1
AWS Layanan terkait	2
Harga	3
Konsep	4
Penagihan dan biaya	5
Memulai	6
Prasyarat	6
Mendaftar untuk Akun AWS	6
Membuat pengguna administratif	7
Siapkan Antarmuka Baris Perintah	8
Menggunakan agen terpadu CloudWatch	8
Menggunakan CloudWatch agen sebelumnya	9
CloudWatch Prasyarat agen log	10
Quick Start: Menginstal agen di instans EC2 Linux yang sedang berjalan	10
Quick Start: Menginstal agen di instans Linux EC2 saat peluncuran	17
Mulai Cepat: Gunakan CloudWatch Log dengan instans Windows Server 2016	21
Mulai Cepat: Gunakan CloudWatch Log dengan instans Windows Server 2012 dan Windows Server 2008	33
Mulai Cepat: Instal agen menggunakan AWS OpsWorks	43
Laporkan status agen CloudWatch Log	49
Mulai agen CloudWatch Log	50
Hentikan agen CloudWatch Log	50
Mulai Cepat dengan AWS CloudFormation	51
Bekerja dengan AWS SDK	53
Menganalisis data log dengan Wawasan CloudWatch Log	55
Memulai: Tutorial kueri	56
Tutorial: Jalankan dan modifikasi kueri sampel	56
Tutorial: Jalankan kueri dengan fungsi agregasi	59
Tutorial: Jalankan kueri yang menghasilkan visualisasi yang dikelompokkan berdasarkan bidang log	60
Tutorial: Jalankan kueri yang menghasilkan visualisasi deret waktu	61
Log yang didukung dan bidang yang ditemukan	62
Bidang di log JSON	64

Sintaks kueri	66
tampilan	68
ladang	68
filter	69
pola	72
mengurai	73
menyortir	75
statistik	75
batasan	81
dedup	82
membuka kedok	82
Boolean, perbandingan, numerik, datetime, dan fungsi lainnya	83
Bidang yang berisi karakter khusus	92
Gunakan alias dan komentar dalam kueri	92
Kueri Sampel	93
Kueri umum	94
Kueri untuk log Lambda	94
Kueri untuk log aliran VPC Amazon	95
Kueri untuk log Route 53	96
Kueri untuk log CloudTrail	96
Pertanyaan untuk Amazon API Gateway	97
Pertanyaan untuk gateway NAT	98
Kueri untuk log server Apache	99
Kueri untuk Amazon EventBridge	100
Contoh perintah parse	100
Visualisasikan data log dalam grafik	101
Simpan dan jalankan kembali kueri	101
Tambahkan kueri ke dasbor atau ekspor hasil kueri	103
Lihat kueri atau riwayat kueri yang sedang berjalan	104
Enkripsi hasil kueri dengan AWS Key Management Service	104
Batas	105
Langkah 1: Buat AWS KMS key	105
Langkah 2: Tetapkan izin pada tombol KMS	106
Langkah 3: Kaitkan kunci KMS dengan hasil kueri Anda	108
Langkah 4: Lepaskan kunci dari hasil kueri di akun	108
Bekerja dengan grup log dan pengaliran log	109

Membuat grup log	109
Mengirim log ke grup log	109
Melihat data log	110
Gunakan Live Tail untuk melihat log dalam waktu dekat	111
Memulai sesi Live Tail	111
Cari data log menggunakan pola filter	113
Cari entri log menggunakan konsol	114
Cari entri log menggunakan AWS CLI	114
Pivot dari metrik ke log	115
Memecahkan masalah	116
Mengubah retensi data log	116
Menandai grup log	117
Dasar tag	118
Melacak biaya menggunakan penandaan	118
Batasan tanda	118
Menandai grup log menggunakan AWS CLI	119
Menandai grup log menggunakan API CloudWatch Log	120
Enkripsi data log menggunakan AWS KMS	120
Batas	121
Langkah 1: Buat AWS KMS kunci	105
Langkah 2: Tetapkan izin pada tombol KMS	106
Langkah 3: Kaitkan kunci KMS dengan grup log	125
Langkah 4: Pisahkan kunci dari grup log	125
Kunci KMS dan konteks enkripsi	126
Membantu melindungi data log sensitif dengan masking	129
Memahami kebijakan perlindungan data	132
Izin IAM diperlukan untuk membuat atau bekerja dengan kebijakan perlindungan data	134
Buat kebijakan perlindungan data seluruh akun	139
Membuat kebijakan perlindungan data untuk satu grup log	143
Lihat data yang dibuka kedoknya	145
Laporan temuan audit	146
Jenis data yang dapat Anda lindungi	148
Filter metrik	188
Konsep	189
Filter sintaks pola untuk filter metrik	190
Mengkonfigurasi nilai metrik untuk filter metrik	191

Menerbitkan dimensi dengan metrik dari peristiwa log	192
Menggunakan nilai dalam peristiwa log untuk menambah nilai metrik	195
Membuat filter metrik	196
Membuat filter metrik untuk grup log	196
Contoh: Hitung peristiwa log	198
Contoh: Hitung kemunculan suatu istilah	199
Contoh: Hitung kode HTTP 404	201
Contoh: Hitung kode HTTP 4xx	203
Contoh: Mengekstraksi bidang dari log Apache dan menetapkan dimensi	205
Daftar filter metrik	207
Menghapus filter metrik	208
Filter langganan	209
Konsep	210
Menggunakan filter langganan	210
Contoh 1: Filter berlangganan dengan Kinesis Data Streams	211
Contoh 2: Filter berlangganan dengan AWS Lambda	217
Contoh 3: Filter berlangganan dengan Amazon Kinesis Data Firehose	220
Berbagi data log lintas akun dengan langganan	227
Berbagi data log lintas akun menggunakan Kinesis Data Streams	228
Berbagi data log lintas akun menggunakan Kinesis Data Firehose	248
Pencegahan Deputi Bingung	262
Sintaks ekspresi filter	263
Sintaks ekspresi reguler	263
Menggunakan ekspresi reguler	266
Ketentuan kecocokan dalam peristiwa log tidak terstruktur	267
Ketentuan kecocokan dalam acara log JSON	271
Memudahkan log acara yang dipisahkan dengan spasi	279
Mengaktifkan logging dari layanan AWS	284
Logging yang membutuhkan izin tambahan [V1]	288
Log dikirim ke CloudWatch Log	289
Log yang dikirim ke Amazon S3	291
Log yang dikirim ke Kinesis Data Firehose	295
Pencatatan yang membutuhkan izin tambahan [V2]	297
Log dikirim ke CloudWatch Log	298
Log yang dikirim ke Amazon S3	300
Log yang dikirim ke Kinesis Data Firehose	304

Pencegahan confused deputy lintas layanan	307
Pembaruan kebijakan	308
Mengekspor data log ke Amazon S3	309
Konsep	310
Ekspor data log ke Amazon S3 menggunakan konsol	311
Ekspor akun yang sama	311
Ekspor lintas akun	318
Ekspor data log ke Amazon S3 menggunakan AWS CLI	326
Ekspor akun yang sama	327
Ekspor lintas akun	334
Jelaskan tugas ekspor	342
Membatalkan tugas ekspor	344
Streaming data ke OpenSearch Layanan	345
Prasyarat	345
Berlangganan grup log ke OpenSearch Layanan	345
Contoh kode	348
Tindakan	349
Kaitkan kunci dengan grup log	349
Membatalkan tugas ekspor	351
Membuat grup log	353
Buat aliran log baru	355
Buat filter langganan	356
Buat tugas ekspor	361
Menghapus grup log	362
Hapus filter langganan	365
Jelaskan filter langganan yang ada	369
Jelaskan tugas ekspor	375
Jelaskan grup log	376
Contoh lintas layanan	379
Menggunakan peristiwa terjadwal untuk menginvokasi fungsi Lambda	380
Keamanan	381
Perlindungan data	382
Enkripsi saat tidak aktif	383
Enkripsi dalam transit	383
Pengelolaan identitas dan akses	383
Autentikasi	383

Kontrol akses	384
Ikhtisar mengenai pengelolaan akses	384
Menggunakan kebijakan berbasis identitas (kebijakan IAM)	390
CloudWatch Referensi izin log	402
Menggunakan peran terkait layanan	408
Validasi kepatuhan	410
Ketahanan	411
Keamanan infrastruktur	411
Titik akhir VPC antarmuka	412
Ketersediaan	412
Membuat titik akhir VPC untuk Log CloudWatch	412
Menguji koneksi antara VPC dan Log CloudWatch	413
Mengontrol akses ke titik akhir VPC CloudWatch Log	413
Support untuk kunci konteks VPC	415
Membuat CloudWatch log panggilan API Amazon LogsAWS CloudTrail	416
CloudWatch Log informasi di CloudTrail	416
Memahami entri file log	418
Referensi agen	420
File konfigurasi agen	420
Menggunakan CloudWatch Agen log dengan proxy HTTP	426
Kompartimentalisasi CloudWatch Log file konfigurasi agen	427
CloudWatch FAQ Log Agent	428
Memantau penggunaan dengan CloudWatch metrik	432
CloudWatch Metrik log	432
Dimensi untuk metrik CloudWatch Logs	436
CloudWatch Log metrik penggunaan layanan	437
Kuota layanan	439
Mengelola kuota layanan CloudWatch Log	445
Riwayat dokumen	447
AWS Glosarium	453

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.

Fitur

- 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 mengirimkan 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 CloudTrail adalah 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, lihat [Membuat CloudWatch log panggilan API Amazon LogsAWS 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.

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 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).

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 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 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](#)
- [Gunakan CloudWatch agen terpadu untuk memulai dengan CloudWatch Log](#)
- [Gunakan CloudWatch agen sebelumnya untuk memulai dengan CloudWatch Log](#)
- [Mulai Cepat: Gunakan AWS CloudFormation untuk memulai dengan CloudWatch Log](#)

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 secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

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

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

Membuat pengguna administratif

Setelah Anda mendaftar Akun AWS, buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di halaman berikutnya, masukkan kata sandi Anda.

Untuk bantuan masuk menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) dalam Panduan Pengguna AWS Sign-In .

2. Aktifkan 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.

Membuat pengguna administratif

- Untuk tugas administratif harian Anda, berikan akses administratif ke pengguna administratif di AWS IAM Identity Center.

Untuk petunjuknya, silakan lihat [Memulai](#) dalam Panduan Pengguna AWS IAM Identity Center .

Masuk sebagai pengguna administratif

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

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.

Dengan 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](#)
- [Mulai Cepat: Instal dan konfigurasikan agen CloudWatch Log pada instans Linux EC2 yang sedang berjalan](#)
- [Mulai Cepat: Instal dan konfigurasikan agen CloudWatch Log pada instans Linux EC2 saat diluncurkan](#)
- [Mulai Cepat: Aktifkan instans Amazon EC2 Anda yang menjalankan Windows Server 2016 untuk mengirim log ke Log menggunakan agen CloudWatch Log CloudWatch](#)
- [Mulai Cepat: Aktifkan instans Amazon EC2 Anda yang menjalankan Windows Server 2012 dan Windows Server 2008 untuk mengirim log ke Log CloudWatch](#)
- [Mulai Cepat: Instal agen CloudWatch Log menggunakan AWS OpsWorks dan Chef](#)
- [Laporkan status agen CloudWatch Log](#)
- [Mulai agen CloudWatch Log](#)
- [Hentikan agen CloudWatch Log](#)

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 instans Linux EC2 yang sedang berjalan

Important

Agen log yang lebih tua 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 mengkonfigurasi 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 [Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya tentang masalah koneksi, lihat [Memecahkan Masalah Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

2. Perbarui instans Amazon Linux Anda untuk mengambil perubahan terbaru dalam repositori paket.

```
sudo yum update -y
```

3. Instalawslogs 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. Hubungkan ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya tentang masalah koneksi, lihat [Memecahkan Masalah Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

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 -0
```

```
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 -0
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/AgentDependencies.tar.gz -0
```

```
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 Wilayah 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-timur-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.
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.

Item	Deskripsi
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 Log 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](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
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](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

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 -0
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, Acara, 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`:

```
"Enabled": 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.

- 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.

- (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.
- (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.

- (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 cap 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.
- (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\\W3SVCn).

 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 cap 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 selengkapnya tentang cara melampirkan IAM role ke instans yang sudah ada, lihat [Melampirkan IAM Role ke Instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Windows.

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 untuk Instans Windows.
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 menampilkan 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 cap 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\\W3SVC1).

 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 cap 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 menggunakan 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 CloudWatch Amazon EC2 untuk Instans Windows](#)

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

bawa 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 Cookbooks 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 instance 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](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
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. Hubungkan ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya tentang masalah koneksi, lihat [Memecahkan Masalah Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux

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. Hubungkan ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya tentang masalah koneksi, lihat [Memecahkan Masalah Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

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. Hubungkan ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya tentang masalah koneksi, lihat [Memecahkan Masalah Menghubungkan ke Instans Anda](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

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": "404Count"  
            }  
        ]  
    }  
}
```

```
        "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 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 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, lihat [Contoh kode untuk CloudWatch Log menggunakan AWS SDK](#).

 Ketersediaan contoh

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

Menganalisis data log dengan Wawasan CloudWatch Log

CloudWatch Logs Insights memungkinkan Anda untuk secara interaktif mencari dan menganalisis data log Anda di Amazon CloudWatch Logs. 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.

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 [StartQuery](#) di Referensi API Amazon CloudWatch Logs.

Daftar Isi

- [Memulai: Tutorial kueri](#)
- [Log yang didukung dan bidang yang ditemukan](#)
- [CloudWatch Sintaks kueri Log Insights](#)
- [Kueri Sampel](#)
- [Visualisasikan data log dalam grafik](#)
- [Simpan dan jalankan kembali kueri CloudWatch Logs Insights](#)
- [Tambahkan kueri ke dasbor atau ekspor hasil kueri](#)
- [Lihat kueri atau riwayat kueri yang sedang berjalan](#)
- [Enkripsi hasil kueri dengan AWS Key Management Service](#)

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, lihat [Memulai 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 CloudWatch log, Wawasan Log secara otomatis mendeteksi bidang data dalam grup. Untuk melihat bidang yang ditemukan, pilih menu Fields di dekat kanan atas halaman.

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.

- 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

- 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.

- (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 atau nilai bidang tertentu. 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. 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 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.

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 menampilkan 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 <code>parse</code> . Untuk informasi selengkapnya, lihat Bidang di log JSON .
CloudTrail log	Untuk informasi selengkapnya, lihat Bidang di log JSON .
Log dalam format 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"  
                }  
            ]  
        }  
    }  
}
```

```
        "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 dengan `"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.

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.

<u>parse</u>	Mengekstrak data dari bidang log untuk membuat bidang yang diekstrak si yang dapat Anda proses dalam kueri Anda. parse mendukung mode glob menggunakan wildcard, dan ekspresi reguler.
<u>sort</u>	Menampilkan peristiwa log yang dikembalikan dalam urutan ascending (asc) atau descending (desc).
<u>stats</u>	Hitung statistik agregat menggunakan nilai di bidang log.
<u>limit</u>	Menentukan jumlah maksimum peristiwa log yang Anda ingin query Anda untuk kembali. sort Berguna dengan mengembalikan hasil “20 teratas” atau “20 terbaru”.
<u>dedup</u>	Menghapus hasil duplikat berdasarkan nilai tertentu di bidang yang Anda tentukan.
<u>unmask</u>	Menampilkan semua konten peristiwa log yang memiliki beberapa konten yang disembunyikan karena kebijakan perlindungan data. Untuk informasi selengkapnya tentang perlindungan data di grup log, lihat Membantu melindungi data log sensitif dengan masking .
<u>Operasi dan fungsi lainnya</u>	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](#)
- [bidang](#)
- [filter](#)
- [pola](#)
- [mengurai](#)
- [menyortir](#)
- [statistik](#)
- [batasan](#)

- [dedup](#)
- [membuka kedok](#)
- [Boolean, perbandingan, numerik, datetime, dan fungsi lainnya](#)
- [Bidang yang berisi karakter khusus](#)
- [Gunakan alias dan komentar dalam kueri](#)

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, dannot).

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 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` `dannot like`, lampirkan substring yang ingin Anda cocokkan dalam tanda kutip tunggal atau ganda. Anda dapat menggunakan pola ekspresi reguler dengan `like` `dannot 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 stringServiceLog.

```
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 stringServiceLog, tetapi juga termasuk stringServiceLog.

```
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 sensitive.

```
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.

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

patternPerintah 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

patternPerintah 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
```

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.

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 `@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”.

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.

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 bin** 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

binFungsi selalu secara implisit menggunakan bidang. @timestamp Ini berarti Anda tidak dapat menggunakan bin 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 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

Fungsi	Tipe hasil	Deskripsi
		lebih rendah dari nilai ini, dan 5 persen lebih tinggi dari nilai ini.
<code>stddev(fieldName: NumericLogField)</code>	nomor	Standar deviasi di bidang yang ditentukan.
<code>sum(fieldName: NumericLogField)</code>	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
<code>earliest(fieldName: LogField)</code>	LogField	Mengembalikan nilai <code>fieldName</code> dari log acara yang memiliki stempel waktu paling awal dalam log yang dikueri.
<code>latest(fieldName: LogField)</code>	LogField	Mengembalikan nilai <code>fieldName</code> dari log acara yang memiliki stempel waktu paling akhir dalam log yang dikueri.
<code>sortsFirst(fieldName: LogField)</code>	LogField	Mengembalikan nilai <code>fieldName</code> yang ada di urutan pertama dalam log yang dikueri.
<code>sortsLast(fieldName: LogField)</code>	LogField	Mengembalikan nilai <code>fieldName</code> 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 dengandedup, 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 dijalankandedup, 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 asc
| 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, lihat [Membantu 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
<code>a + b</code>	Penambahan
<code>a - b</code>	Pengurangan
<code>a * b</code>	Perkalian
<code>a / b</code>	Pembagian
<code>a ^ b</code>	Eksponensiasi (pengembalian) <code>2 ^ 3 8</code>
<code>a % b</code>	Sisa atau modulus (pengembalian) <code>10 % 3 1</code>

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
<code>abs(a: number)</code>	nomor	Nilai absolut

Operasi	Tipe Hasil	Deskripsi
<code>ceil(a: number)</code>	nomor	Bulat ke langit-langit (bilangan bulat terkecil yang lebih besar dari nilai) a
<code>floor(a: number)</code>	nomor	Bulat ke lantai (bilangan bulat terbesar yang lebih kecil dari nilai) a
<code>greatest(a: number, ...numbers: number[])</code>	nomor	Mengembalikan nilai terbesar
<code>least(a: number, ...numbers: number[])</code>	nomor	Mengembalikan nilai terkecil
<code>log(a: number)</code>	nomor	Log alami
<code>sqrt(a: number)</code>	nomor	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 `m` selama beberapa menit atau `h` berjam-jam. Misalnya, `10m` adalah 10 menit, dan `1h` 1 jam. 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
<code>bin(period: Period)</code>	Stempel Waktu	<p>Membulatkan nilai @timestamp ke periode waktu tertentu dan kemudian memotong. Misalnya, <code>bin(5m)</code> bulatkan nilai @timestamp ke 5 menit terdekat.</p> <p>Anda dapat menggunakan ini untuk mengelompokkan beberapa entri log bersama-sama dalam kueri. Contoh berikut mengembalikan jumlah pengecualian per jam:</p> <pre>filter @message like /Exception/ stats count(*) as exceptionCount by bin(1h) sort exceptionCount desc</pre> <p>Satuan waktu dan singkatan berikut didukung dengan <code>bin</code> fungsi tersebut. Untuk semua unit dan singkatan yang menyertakan lebih dari satu karakter, menambahkan s ke pluralisasi didukung. Jadi keduanya <code>hr</code> dan <code>hrs</code> bekerja untuk menentukan jam.</p> <ul style="list-style-type: none"> • <code>millisecond ms msec</code> • <code>second s sec</code> • <code>minute m min</code> • <code>hour h hr</code> • <code>day d</code> • <code>week w</code> • <code>month mo mon</code> • <code>quarter q qtr</code>

Fungsi	Tipe hasil	Deskripsi
		<ul style="list-style-type: none"> • <code>year</code> y yr
<code>datefloor(timestamp : Timestamp, period : Period)</code>	Stempel Waktu	Memotong stempel waktu ke periode tertentu. Misalnya, <code>datefloor(@timestamp, 1h)</code> memotong semua nilai <code>@timestamp</code> ke bagian bawah jam.
<code>dateceil(timestamp : Timestamp, period : Period)</code>	Stempel waktu	Membulatkan stempel waktu ke periode tertentu dan kemudian memotong. Misalnya, <code>dateceil(@timestamp, 1h)</code> memotong semua nilai <code>@timestamp</code> ke bagian atas jam.
<code>fromMillis(fieldName : number)</code>	Stempel waktu	Menafsirkan bidang input sebagai jumlah miliditik sejak jangka waktu Unix dan mengubahnya menjadi stempel waktu.
<code>toMillis(fieldName : Timestamp)</code>	nomor	Mengonversi stempel waktu yang ditemukan di bidang bernama menjadi angka yang mewakili miliditik sejak jangka waktu Unix. Misalnya, <code>toMillis(@timestamp)</code> mengubah stempel waktu <code>2022-01-14T13:18:031.000-08:00</code> menjadi <code>1642195111000</code>

 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
<code>ispresent(fieldName: LogField)</code>	Boolean	Mengembalikan <code>true</code> jika bidang ada
<code>coalesce(fieldName: LogField, ...fieldNames: LogField[])</code>	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
<code>isValidIp(fieldName: string)</code>	boolean	Mengembalikan <code>true</code> jika bidang adalah alamat IPv4 atau IPv6 yang valid.
<code>isValidIpv4(fieldName: string)</code>	boolean	Mengembalikan <code>true</code> jika bidang adalah alamat IPv4 yang valid.
<code>isValidIpv6(fieldName: string)</code>	boolean	Mengembalikan <code>true</code> jika bidang adalah alamat IPv6 yang valid.
<code>isIpInSubnet(fieldName: string, subnet: string)</code>	boolean	Mengembalikan <code>true</code> jika bidang adalah alamat IPv4 atau IPv6 yang valid dengan subnet v4 atau v6 yang ditentukan. Saat Anda menentukan subnet, gunakan notasi CIDR seperti <code>192.0.2.0/24</code> atau <code>2001:db8::/32</code> , di mana <code>192.0.2.0</code> atau <code>2001:db8::</code> merupakan awal dari blok CIDR.

Fungsi	Tipe hasil	Deskripsi
<code>isIPv4InSubnet(fileIdName: string, subnet: string)</code>	boolean	Mengembalikan <code>true</code> jika bidang adalah alamat IPv4 yang valid dalam subnet v4 yang ditentukan. Saat Anda menentukan subnet, gunakan notasi CIDR seperti <code>192.0.2.0/24</code> di <code>192.0.2.0</code> mana awal blok CIDR..
<code>isIPv6InSubnet(fileIdName: string, subnet: string)</code>	boolean	Mengembalikan <code>true</code> jika bidang adalah alamat IPv6 yang valid dalam subnet v6 yang ditentukan. Saat Anda menentukan subnet, gunakan notasi CIDR seperti <code>2001:db8::/32</code> di <code>2001:db8::</code> 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
<code>isempty(fieldName: string)</code>	Angka	Mengembalikan <code>1</code> jika bidang tidak ada atau string kosong.
<code>isblank(fieldName: string)</code>	Angka	Mengembalikan <code>1</code> jika bidang tidak ada, string kosong, atau hanya berisi spasi.
<code>concat(str: string, ...strings: string[])</code>	string	Menyatukan string.
<code>ltrim(str: string)</code>	string	Jika fungsi tidak memiliki argumen kedua, ia menghapus spasi putih dari kiri string. Jika fungsi memiliki argumen string kedua, itu
<code>ltrim(str: string, trimChars: string)</code>		

Fungsi	Tipe hasil	Deskripsi
		tidak menghapus spasi putih. Sebaliknya, ia menghapus karakter <code>trimChars</code> dari <code>kiristr</code> . Misalnya, <code>ltrim("xyZxyfooxyz", "xyz")</code> mengembalikan "fooxyz".
<code>rtrim(str: string)</code> <code>rtrim(str: string, trimChars: string)</code>	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 <code>trimChars</code> dari dari kanan <code>str</code> . Misalnya, <code>rtrim("xyzfooxyxyz", "xyz")</code> mengembalikan "xyzfoo".
<code>trim(str: string)</code> <code>trim(str: string, trimChars: string)</code>	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 <code>trimChars</code> dari kedua sisistr. Misalnya, <code>trim("xyzxyfooxyz", "xyz")</code> mengembalikan "foo".

Fungsi	Tipe hasil	Deskripsi
<code>strlen(str: string)</code>	nomor	Mengembalikan panjang string dalam poin kode Unicode.
<code>toupper(str: string)</code>	string	Mengonversi string menjadi huruf besar.
<code>tolower(str: string)</code>	string	Mengonversi string menjadi huruf kecil.
<code>substr(str: string, startIndex: number)</code> <code>substr(str: string, startIndex: number, length: number)</code>	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, <code>substr("xyZfooxyZ", 3, 3)</code> mengembalikan "foo".
<code>replace(fieldName: string, searchValue: string, replaceValue: string)</code>	string	Mengganti semua <code>searchValue</code> dalam <code>fieldName: string</code> dengan <code>replaceValue</code> . Misalnya, fungsi <code>replace(logGroup, "smoke_test", "Smoke")</code> mencari peristiwa log di mana bidang <code>logGroup</code> berisi nilai string <code>smoke_test</code> dan menggantikan nilai dengan string <code>Smoke</code> .
<code>strcontains(str: string, searchValue: string)</code>	nomor	Mengembalikan 1 jika <code>str</code> berisi <code>searchValue</code> dan 0 sebaliknya.

Bidang yang berisi karakter khusus

Anda harus mengelilingi bidang log yang dinamai dalam kueri yang menyertakan karakter selain @ simbol, periode (.), dan karakter non-alfanumerik dalam tombol 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 `aliasID`.

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 aliastime.

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
```

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.

Topik

- [Kueri umum](#)
- [Kueri untuk log Lambda](#)
- [Kueri untuk log aliran VPC Amazon](#)
- [Kueri untuk log Route 53](#)
- [Kueri untuk log CloudTrail](#)
- [Pertanyaan untuk Amazon API Gateway](#)
- [Pertanyaan untuk gateway NAT](#)
- [Kueri untuk log server Apache](#)
- [Kueri untuk Amazon EventBridge](#)
- [Contoh perintah parse](#)

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 provisionedMemoryMB,  
min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,  
avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,  
max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,  
provisionedMemoryMB - 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 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 mengurnya. Anda dapat menyimpan sebanyak 1000 kueri per wilayah dan per akun.

Untuk menyimpan kueri, Anda harus masuk ke peran yang memiliki izinlogs:PutQueryDefinition. Untuk melihat daftar kueri yang disimpan, Anda harus masuk ke peran yang memiliki izinlogs: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. Di 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 menggunakan 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-policy](#) perintah 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](#).

ConditionBagian 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-policy](#) perintah 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-key](#) perintah 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:*
```

Bekerja dengan grup log dan pengaliran log

Pengaliran log adalah urutan log acara yang berbagi sumber yang sama. Setiap sumber log 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. Di 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 cursor ke grup log yang ingin Anda favoritkan, 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 CLI[put-log-events](#)—Mengunggah kumpulan peristiwa log ke Log. CloudWatch
- Secara terprogram - [PutLogEvents](#)API memungkinkan Anda untuk mengunggah batch peristiwa log secara terprogram ke Log. CloudWatch

Lihat data log yang dikirim ke CloudWatch Log

Anda dapat melihat dan mengulir 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. Di 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](#).

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, lihat [Membantu 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, Live tail.
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 diAdd 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. (Optional) 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. Di 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. Di 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-events](#) perintah 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-events perintah 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.

Memecahkan 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 Grup log.
3. Temukan grup log yang akan diperbarui.
4. Di kolom Expire Events After (Akhiri Acara Setelah) untuk grup log tersebut, pilih pengaturan retensi saat ini, seperti Never Expire (Tidak Pernah Berakhir).
5. Di Edit Retention (Edit Retensi), untuk Retention (Retensi), pilih nilai retensi log, lalu pilih Ok.

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 penugasan.

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 tag](#)
- [Melacak biaya menggunakan penandaan](#)
- [Batasan tanda](#)
- [Menandai grup log menggunakan AWS CLI](#)

- [Menandai grup log menggunakan API CloudWatch Log](#)

Dasar tag

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 tanda

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 kuncinya 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-policy](#) perintah 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-policy](#) perintah 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-groups](#) perintah 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-group](#) perintah 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-key](#) perintah 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.

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 Perlindungan data log memungkinkan Anda memanfaatkan pencocokan pola dan model pembelajaran mesin untuk mendeteksi data sensitif. Kriteria dan teknik yang digunakan disebut sebagai pengidentifikasi data terkelola. Teknik ini dapat mendeteksi daftar besar tipe data sensitif untuk banyak negara dan wilayah, termasuk data keuangan, informasi identitas pribadi, dan informasi kesehatan yang dilindungi. Untuk beberapa jenis data, deteksi tergantung pada juga menemukan kata kunci tertentu yang berdekatan dengan data sensitif.

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, lihat [Memantau 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, lihat [Jenis data yang dapat Anda lindungi](#).

Daftar Isi

- [Memahami kebijakan perlindungan data](#)
 - [Apa itu kebijakan perlindungan data?](#)
 - [Bagaimana kebijakan perlindungan data terstruktur?](#)
 - [Properti JSON untuk kebijakan perlindungan data](#)
 - [Properti JSON untuk pernyataan kebijakan](#)
 - [Properti JSON untuk operasi pernyataan kebijakan](#)
- [Izin IAM diperlukan untuk membuat atau bekerja dengan kebijakan perlindungan data](#)
 - [Izin yang diperlukan untuk kebijakan perlindungan data tingkat akun](#)
 - [Izin yang diperlukan untuk kebijakan perlindungan data untuk satu grup log](#)
 - [Contoh kebijakan perlindungan data](#)
- [Buat kebijakan perlindungan data seluruh akun](#)
 - [Konsol](#)
 - [AWS CLI](#)
 - [Sintaks kebijakan perlindungan data untuk AWS CLI atau operasi API](#)
- [Membuat kebijakan perlindungan data untuk satu grup log](#)
 - [Konsol](#)
 - [AWS CLI](#)
 - [Sintaks kebijakan perlindungan data untuk AWS CLI atau operasi API](#)
- [Lihat data yang dibuka kedoknya](#)
- [Laporan temuan audit](#)
 - [Kebijakan kunci yang diperlukan untuk mengirim temuan audit ke emer yang dilindungi oleh AWS KMS](#)
- [Jenis data yang dapat Anda lindungi](#)
 - [CloudWatch Pengidentifikasi data terkelola log untuk tipe data sensitif](#)
 - [Kredensial](#)
 - [ARN pengenal data untuk tipe data kredensi](#)
 - [Pengidentifikasi perangkat](#)
 - [ARN pengenal data untuk tipe data perangkat](#)

- [Informasi keuangan](#)
 - [ARN pengenal data untuk tipe data keuangan](#)
- [Informasi kesehatan yang dilindungi \(PHI\)](#)
 - [ARN pengidentifikasi data untuk tipe data informasi kesehatan yang dilindungi \(PHI\)](#)
- [Informasi Identifikasi Pribadi \(PII\)](#)
 - [Kata kunci untuk nomor identifikasi surat izin mengemudi](#)
 - [Kata kunci untuk nomor induk kependudukan](#)
 - [Kata kunci untuk nomor paspor](#)
 - [Kata kunci untuk nomor pokok wajib pajak](#)
 - [ARN pengidentifikasi data untuk informasi identitas pribadi \(PII\)](#)

Memahami kebijakan perlindungan data

Topik

- [Apa itu kebijakan perlindungan data?](#)
- [Bagaimana kebijakan perlindungan data terstruktur?](#)

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, lihat[Laporan temuan audit](#).

Untuk informasi selengkapnya tentang metrik yang dikirimkan oleh CloudWatch Log CloudWatch, lihat[Memantau dengan CloudWatch metrik](#).

- De-identifikasi - Tutupi data sensitif tanpa mengganggu logging.

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	*
Membuat kebijakan perlindungan data dengan Kinesis Data Firehose sebagai tujuan audit	logs:CreateLogDelivery	*
	logs:PutResourcePolicy	*
Membuat kebijakan perlindungan data dengan Kinesis Data Firehose sebagai tujuan audit	logs:DescribeResourcePolicies	*
	logs:DescribeLogGroups	*
Membuat kebijakan perlindungan data dengan Kinesis Data Firehose sebagai tujuan audit	logs:PutAccountPolicy	*
	logs:PutDataProtectionPolicy	*
Membuat kebijakan perlindungan data dengan Kinesis Data Firehose sebagai tujuan audit	logs:CreateLogDelivery	*
	firehose:TagDeliveryStream	arn:aws:logs:::deliverystream/ <i>YOUR_DELIVERY_STREAM</i>

Operasi	Izin IAM diperlukan	Sumber daya
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:*
Melihat kebijakan perlindungan data yang ada	logs:GetDataProtectionPolicy	*
Menghapus kebijakan perlindungan data	logs>DeleteAccountPolicy logs>DeleteDataProtectionPolicy	*
	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: <i>YOUR_LOG_GROUP</i> :*
Membuat kebijakan perlindungan data dengan CloudWatch Log sebagai tujuan audit	logs:PutDataProtectionPolicy logs>CreateLogDelivery logs:PutResourcePolicy logs:DescribeResourcePolicies logs:DescribeLogGroups	arn:aws:logs:::log-group: <i>YOUR_LOG_GROUP</i> :* * * *
Membuat kebijakan perlindungan data dengan Kinesis Data Firehose sebagai tujuan audit	logs:PutDataProtectionPolicy logs>CreateLogDelivery firehose:TagDeliveryStream	arn:aws:logs:::log-group: <i>YOUR_LOG_GROUP</i> :* * arn:aws:logs:::deliverystream/ <i>YOUR_DELIVERY_STREAM</i>

Operasi	Izin IAM diperlukan	Sumber daya
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 :*
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 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, 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, lihat [Jenis data yang dapat Anda lindungi](#).

6. (Opsional) Pilih satu atau beberapa 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.
7. 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 `DataIdentifier` array dan `Operation` properti dengan `Audit` tindakan. `DataIdentifier` 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 Kinesis Data Firehose, dan bucket S3, mereka harus sudah ada. Untuk contoh laporan audit findins, lihat [Laporan temuan audit](#)

- Blok kedua harus menyertakan `DataIdentifier` array dan `Operation` properti dengan `Deidentify` tindakan. `DataIdentifier` array harus sama persis dengan `DataIdentifier` 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": {}  
        }  
    }  
}  
]  
}
```

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, 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, lihat[Jenis data yang dapat Anda lindungi](#).

6. (Opsional) Pilih satu atau beberapa 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.
7. 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 `DataIdentifier` array dan `Operation` properti dengan `Audit` tindakan. `DataIdentifier` 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 Kinesis Data Firehose, dan bucket S3, mereka harus sudah ada. Untuk contoh laporan audit findins, lihat [Laporan temuan audit](#)

- Blok kedua harus menyertakan `DataIdentifier` array dan `Operation` properti dengan `Deidentify` tindakan. `DataIdentifier` array harus sama persis dengan `DataIdentifier` 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",  
  "DataIdentifiers": [  
    {"Type": "Email", "Value": "john.doe@example.com"},  
    {"Type": "SIM", "Value": "12345678901234567890"}],  
  "Operations": [  
    {"Action": "Audit", "Details": {"FindingsDestination": {"Type": "Log", "LogGroup": "my-log-group", "Region": "us-west-2"}}, "Type": "Audit"},  
    {"Action": "Deidentify", "Details": {"MaskConfig": {}, "Type": "Deidentify"}},  
    {"Action": "Audit", "Details": {"FindingsDestination": {"Type": "Log", "LogGroup": "my-log-group", "Region": "us-west-2"}}, "Type": "Audit"}]  
}
```

```
"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, lihat [CloudWatch Sintaks kueri Log Insights](#).

- Gunakan [GetLogEvents](#) atau [FilterLogEvents](#) operasi dengan unmask parameter.

CloudWatchLogsFullAccessKebijakan tersebut termasuk logs : Unmask izin. Untuk memberikan logs : Unmask kepada pengguna yang tidak memiliki CloudWatchLogsFullAccess, Anda dapat melampirkan kebijakan IAM kustom ke 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 Kinesis Data 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 sensitif 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, dan negara dan wilayah mana yang relevan untuk setiap jenis data.

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 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.

Daftar Isi

- [CloudWatch Pengidentifikasi data terkelola log untuk tipe data sensitif](#)
- [Kredensial](#)
 - [ARN pengenal data untuk tipe data kredensi](#)
- [Pengidentifikasi perangkat](#)
 - [ARN pengenal data untuk tipe data perangkat](#)
- [Informasi keuangan](#)
 - [ARN pengenal data untuk tipe data keuangan](#)
- [Informasi kesehatan yang dilindungi \(PHI\)](#)
 - [ARN pengidentifikasi data untuk tipe data informasi kesehatan yang dilindungi \(PHI\)](#)
- [Informasi Identifikasi Pribadi \(PII\)](#)
 - [Kata kunci untuk nomor identifikasi surat izin mengemudi](#)
 - [Kata kunci untuk nomor induk kependudukan](#)
 - [Kata kunci untuk nomor paspor](#)
 - [Kata kunci untuk nomor pokok wajib pajak](#)
 - [ARN pengidentifikasi data untuk informasi identitas pribadi \(PII\)](#)

CloudWatch Pengidentifikasi data terkelola log untuk tipe data sensitif

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
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
IndividualTaxIdentificationNumber	Pribadi	AS
InseeCode	Pribadi	FR
MedicareBeneficiaryNumber	Kondisi	AS
NationalDrugCode	Kondisi	AS
NationalIdentificationNumber	Pribadi	DE, ES, ITU

Pengidentifikasi	Kategori	Negara dan bahasa
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
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-awscredentials	Semua
Kunci pribadi OpenSSH	OpenSSHPri vateKey	Tidak ada	Semua
Kunci pribadi PGP	PgpPrivateKey	Tidak ada	Semua
Kunci Pribadi PKCS	PkcsPriva teKey	Tidak ada	Semua
Kunci pribadi PuTTY	PuttyPriv ateKey	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: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	Prancis, Jerman, Italia, Spanyol, Inggris	Termasuk Internasional Bank Account

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
		rekening bank nanti di bagian ini.		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	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
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 club, discover, electron, japanese card bureau, jcb, mastercard, mc, pan, payment account number, payment card number, pcn, union pay, visa	Semua	Deteksi mengharus kan data menjadi urutan 13-19 digit yang mematuhi rumus pemeriksa an Luhn, dan menggunakan awalan nomor kartu standar untuk salah satu jenis kartu kredit berikut: American Express, Dankort, Diner's

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
				Club, Discover, Electron, Japanese Card Bureau (JCB), Mastercar d, dan Visa. UnionPay
Kode verifikasi kartu kredit	CreditCar dSecurity Code	card id, card identification code, card identific ation number , card security code, card validation code , card validatio n number , card verification data , card verification value, cvc, cvc2, cvv, cvv2, elo verificat ion 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, kartenummer , kontonummer , kreditkartennummer , sepa
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

CloudWatch Log tidak melaporkan kejadian urutan berikut, yang telah disediakan oleh penerbit kartu kredit untuk pengujian publik.

```
12200000000003, 2222405343248877, 2222990905257051, 2223007648726984,
2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
36148900647913,
36700102000000, 371449635398431, 37828224631005, 378734493671000, 3852000023237,
401288888881881,
4111111111111111, 422222222222, 44443332221111, 4462030000000000, 4484070000000000,
491183000000,
```

```
4917300800000000, 4917610000000000, 491761000000000003, 5019717010103742,  
5105105105100,  
5111010030175156, 5185540810000019, 52008282828210, 5204230080000017,  
5204740009900014, 5420923878724339,  
5454545454545454, 5455330760000018, 5506900490000436, 5506900490000444,  
5506900510000234, 5506920809243667,  
5506922400634930, 5506927427317625, 5553042241984105, 555553753048194,  
555555555554444, 5610591081018250,  
6011000990139424, 6011000400000000, 601111111111117, 630490017740292441,  
630495060000000000,  
6331101999990016, 6759649826438453, 679999010000000019, 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)	DrugEnforcementAge ncyNumber	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,	Uni Eropa

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah
		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 de seguro, sairaanhoitokortin , sairausvakuutusnumero , sjukförsäkring nummer, 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

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah
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
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	Canada

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 pengidentifikasi data PHI

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: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	Sebaran	Support mencakup sebagian besar format tanggal, 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	Brazil	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
		endereçamento postal		
Kadastro Nacional da Pessoa Jurídica (CNPJ)	Cnpj	cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj	Brazil	
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	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
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	
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	

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
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).
Nomor Asuransi Nasional (NINO)	NationalInsuranceNumber	insurance no., insurance number, insurance# , national insurance number, nationalinsurance# , nationalinsurance#, nin, nino	–	Britania Raya

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
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	Canada	

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,phc number,tel,telephone , telephone number	Brasil, Kanada, Prancis, Jerman, Italia, Spanyol, Inggris, 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	Ini termasuk nomor bebas pulsa Amerika Serikat dan nomor faks.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
				data, nomor tersebut harus menyertakan kode negara.
Kode Pos	PostalCode	Tidak ada	Canada	
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	Canada	

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,social security no# ,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	Sebagian	

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	Sebaran	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° 56'55.010 4 "N 87° 39'19.119 6"W.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nama lengkap	Name	Tidak ada	Sebaran	CloudWatc h Log hanya dapat mendeteks i nama lengkap. Dukungan terbatas pada set karakter Latin.

Jenis data	ID pengenal data	Diperlukan kata kunci	Negara dan wilayah	Catatan
Nomor Identifikasi Kendaraan (VIN)	VehicleId entificationNumber	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	Sebaran	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, führerscheinnummer, fuhrerschein, führerschein, führerschein- nr, führerschein- nr, führerscheinnummer, 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, císla licence řidiče, číslo řidičského průkazu, ovladače lic., povolení k jízdě, povolení řidiče, řidiči povolení, řidičský průkaz, 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, fuehrerscheinnummer, fuhrerschein, führerschein, führerschein- nr, führerschein- nr, führerscheinnummer, führerscheinnummer, 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	licenzja 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 conductor, el carnet de conductor, licencia conductor, licencia de manejo, número carnet conductor, número de carnet de conductor, número de permiso conductor, número de permiso de conductor, número licencia conductor, número permiso conductor, permiso conducción, permiso conductor, 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'identité, 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#, nationalidno#, 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, reisepassnr, 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-R0

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

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.

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

- [Konsep](#)
- [Filter sintaks pola untuk filter metrik](#)
- [Membuat filter metrik](#)
- [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, lihat [CloudWatch 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 pencocokan dalam peristiwa log tidak terstruktur
- Ketentuan yang cocok dalam peristiwa log JSON
- Istilah yang cocok 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 [MetricDatum](#)di Referensi Amazon CloudWatch API.

Topik

- [Mengkonfigurasi nilai metrik untuk filter metrik](#)
- [Menerbitkan dimensi dengan metrik dari nilai di JSON atau peristiwa log yang dibatasi ruang](#)
- [Menggunakan nilai dalam peristiwa log untuk menambah nilai metrik](#)

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 bernaama, 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 biaya](#).

Mempublikasikan 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 namaserver, dan nilai dimensi dalam peristiwa log contoh adalah "Prod".

Example: Match terms with AND (&&) and OR (||)

Anda dapat menggunakan operator logis 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. Di 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. Di 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 Metric Details (Detail Metrik), untuk Metric Namespace, 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 penggugah/prompt 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. Di 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 **ApacheNotFoundCount**.
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=ApacheNotFoundCount,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. Di 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.

Aturan ekstraksi ini cocok dengan tujuh bidang log acara. Nilai metrik adalah nilai token ketujuh yang cocok. Anda dapat melihat referensi ke token sebagai "\$7" di bidang `metricValue` aturan ekstraksi.

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. Di 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='$size',unit=Bytes,dimensions=$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. Di 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. Di 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 Kinesis Data Firehose, atau untuk pemrosesan, analisis, atau AWS Lambda 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.

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 mengirim ulang jika kesalahan adalah kesalahan yang tidak dapat dicoba ulang, seperti atau. AccessDeniedException ResourceNotFoundException

CloudWatch Log juga menghasilkan CloudWatch metrik tentang penerusan peristiwa log ke langganan. Untuk informasi selengkapnya, lihat [Memantau 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](#).

Daftar Isi

- [Konsep](#)
- [Menggunakan filter langganan CloudWatch Log](#)
- [Berbagi data log lintas akun dengan langganan](#)
- [Pencegahan Deputi Bingung](#)

Konsep

Setiap filter langganan terdiri dari elemen kunci berikut:

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.

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 Kinesis Data Streams, aliran Kinesis Data 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.

Menggunakan filter langganan CloudWatch Log

Anda dapat menggunakan filter langganan dengan Kinesis Data Streams, Lambda, atau Kinesis Data 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

- [Contoh 1: Filter berlangganan dengan Kinesis Data Streams](#)
- [Contoh 2: Filter berlangganan dengan AWS Lambda](#)
- [Contoh 3: Filter berlangganan dengan Amazon Kinesis 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 kredensial "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 dijatuahkan.

Untuk mengurangi risiko pelambatan, Anda dapat mengambil langkah-langkah berikut:

- 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 [Memantau dengan CloudWatch metrik](#).
- Gunakan mode kapasitas sesuai permintaan untuk streaming Anda di Kinesis Data Streams. Mode sesuai permintaan langsung mengakomodasi beban kerja Anda saat 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 berlangganan 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 `RoleArn` 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": "AA0IIAH450GAB4HC5F431",  
    "CreateDate": "2023-01-12T12:00:00Z",  
    "LastUsed": "2023-01-12T12:00:00Z",  
    "ARN": "arn:aws:iam::123456789012:role/RoleName",  
    "Path": "/",  
    "MaxSessionDuration": 3600,  
    "Description": "A temporary role for CloudWatch Logs.",  
    "AssumeRolePolicy": {  
      "Version": "2012-10-17",  
      "Statement": [ { "Action": "sts:AssumeRole", "Effect": "Allow", "Principal": "logs.amazonaws.com" } ]  
    },  
    "PermissionsBoundary": "arn:aws:iam::123456789012:policy/CloudWatchLogsFullAccess",  
    "Tags": [ { "Key": "Name", "Value": "RoleName" } ]  
  }  
}
```

```

        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}

```

- 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"
    }
  ]
}
```

- Kaitkan kebijakan izin dengan peran menggunakan [put-role-policy](#) perintah berikut:

```
aws iam put-role-policy --role-name CWLtoKinesisRole --policy-name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL-Kinesis.json
```

- 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"
```

- 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/  
        kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL  
        +wev+e2P4djJg4L9wmXKvQYoE+rMUiFq  
        +p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+0IK8zM5My8ID  
        +g6rMo7UKWeI4+IWik20Sh0uP"  
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/  
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL  
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq  
+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+0IK8zM5My8ID  
+q6rMo7UKWeI4+IWiK20Sh0uP"
```

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": "1",  
      "time": "2018-01-01T00:00:00Z",  
      "type": "Data",  
      "data": {  
        "version": "1.0",  
        "type": "File",  
        "fileName": "file1.log",  
        "fileSize": 1000,  
        "fileContent": "File content 1"  
      }  
    },  
    {  
      "id": "2",  
      "time": "2018-01-01T00:00:01Z",  
      "type": "Data",  
      "data": {  
        "version": "1.0",  
        "type": "File",  
        "fileName": "file2.log",  
        "fileSize": 2000,  
        "fileContent": "File content 2"  
      }  
    }  
  ]  
}
```

```
        "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 didekod 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 Kinesis 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 Kinesis Data Firehose. Data yang dikirim dari CloudWatch Log ke Amazon Kinesis Data Firehose sudah dikompresi dengan kompresi gzip level 6, sehingga Anda tidak perlu menggunakan kompresi dalam aliran pengiriman Kinesis Data Firehose Anda.

 Note

Sebelum membuat pengaliran Kinesis Data Firehose, hitung volume data log yang akan dihasilkan. Pastikan untuk membuat pengaliran Kinesis Data Firehose yang dapat menangani volume ini. Jika pengaliran tidak dapat menangani volume, pengaliran log akan mengalami

throttling. Untuk informasi selengkapnya tentang batas volume pengaliran Kinesis Data Firehose, lihat [Batas Data Amazon Kinesis Data Firehose](#).

Untuk membuat filter berlangganan untuk Kinesis Data 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 Kinesis Data Firehose untuk memasukkan data ke dalam bucket Amazon S3 Anda.

Untuk informasi selengkapnya, lihat [Mengontrol Akses dengan Amazon Kinesis Data Firehose](#) dalam Panduan Developer Amazon Kinesis 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": "AA0IIAH450GAB4HC5F431",
    "CreateDate": "2015-05-29T13:46:29.431Z",
    "RoleName": "FirehosetoS3Role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
  }
}
```

4. Buat kebijakan izin untuk menentukan tindakan yang dapat dilakukan oleh Kinesis Data 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:::logdata-*"
    }
  ]
}
```

```

    "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 Kinesis Data Firehose sebagai berikut, dengan mengganti 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 Kinesis Data Firehose secara otomatis menggunakan prefiks 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 berakhiri 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 perintah Kinesis describe-delivery-streamData 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 dalam aliran pengiriman Firehose Data Kinesis 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 Kinesis Data Firehose dalam status aktif dan Anda telah membuat peran IAM, Anda dapat CloudWatch membuat filter langganan Log. Filter langganan segera memulai aliran data log nyata dari grup log yang dipilih ke aliran pengiriman Amazon Kinesis 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 Kinesis Data Firehose Anda. Data Anda akan mulai muncul di Amazon S3 berdasarkan interval buffer waktu yang ditetapkan pada aliran pengiriman Amazon Kinesis 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
```

Berbagi data log lintas akun dengan langganan

Anda dapat berkolaborasi dengan pemilik AWS akun lain dan menerima peristiwa log mereka di AWS sumber daya Anda, seperti Amazon Kinesis atau aliran Amazon Kinesis Data Firehose (ini dikenal sebagai berbagi data lintas akun). Misalnya, data peristiwa log ini dapat dibaca dari Aliran

Data Kinesis terpusat atau aliran Firehose Data Kinesis 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.

Topik

- [Berbagi data log lintas akun menggunakan Kinesis Data Streams](#)
- [Berbagi data log lintas akun menggunakan Kinesis Data Firehose](#)

Berbagi data log 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 sisa bagian ini, pengirim data log ditampilkan dengan nomor AWS akun fiks 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 fiks 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.

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](#)
- [Memperbarui langganan lintas akun yang ada](#)

Menyiapkan langganan lintas akun baru

Ikuti langkah-langkah di bagian ini untuk menyiapkan langganan log lintas akun baru.

Topik

- [Langkah 1: Buat tujuan](#)
- [Langkah 2: \(Hanya jika menggunakan organisasi\) Buat peran IAM](#)
- [Langkah 3: Tambah/validasi izin IAM untuk tujuan lintas akun](#)
- [Langkah 4: Buat filter berlangganan](#)

- [Validasi alur peristiwa log](#)
- [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": "34028236692093846346337460743176EXAMPLE",
        "StartingHashKey": "0"
    },
    "SequenceNumberRange": {
        "StartingSequenceNumber":
            "4955113521868881845667950383198145878459135270218EXAMPLE"
    }
}
]
}
}

```

Mungkin diperlukan satu atau dua menit bagi pengaliran Anda untuk muncul dalam keadaan aktif.

- Buat peran IAM yang memberikan izin kepada CloudWatch Log untuk memasukkan data ke aliran Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file ~/TrustPolicyForCWL.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": "AA0IIAH450GAB4HC5F431",
            "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 ~/PermissionsForCWL.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 DestinationArn yang 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 (11111111111) 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 ke [Langkah 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 berlangganan. 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. PerhatikanArn 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 ke [Langkah 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 Kinesis Data 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 ke [Langkah 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, akun123456789012:

```
{  
    "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, akun123456789012:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Allow subscription filters on one specific resource in one  
            specific account",  
            "Effect": "Allow",  
            "Resource": "arn:aws:logs:123456789012:destination:sampleDestination"  
        }  
    ]  
}
```

```
        "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 kredensial “Root” dikirim 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.

Ketika Anda memasukkan perintah berikut, pastikan Anda masuk sebagai pengguna IAM atau menggunakan peran IAM yang Anda tambahkan kebijakan untuk, in. [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": "AAAAAAAFAFGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
  --limit 10 \
  --shard-iterator
  "AAAAAAAFAFGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+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 didekripsi 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": "3195310660696698337880902507980421114328961542429EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}}"
        },
        {
            "id": "3195310660696698337880902507980421114328961542429EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}}"
        },
        {
            "id": "3195310660696698337880902507980421114328961542429EXAMPLE",
            "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 11111111111 yang ditambahkan sudah sebelumnya dihentikan dari mengirim data log lagi, dan akun 22222222222 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 11111111111 dihentikan, dan akun 22222222222 diaktifkan. Letakkan kebijakan ini di file ~/ NewAccessPolicy.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "22222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Panggilan PutDestinationPolicy untuk 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 langganan](#)
- [Langkah 2: Perbarui kebijakan akses tujuan yang ada](#)

Langkah 1: Perbarui filter langganan

Note

Langkah ini diperlukan hanya untuk langganan lintas akun untuk log yang dibuat oleh layanan yang tercantum di. [Mengaktifkan logging dari layanan AWS](#) 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 akun 999999999999. 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 namatestDestination.

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 [Mengaktifkan logging dari layanan AWS](#), maka sebelum melakukan langkah ini, Anda harus terlebih dahulu memperbarui filter langganan di semua akun pengirim seperti yang dijelaskan di [Langkah 1: Perbarui filter langganan](#).

```
aws logs put-destination-policy  
  \ --destination-name "testDestination"
```

```
\ --access-policy file://~/AccessPolicy.json  
\ --force
```

Berbagi data log lintas akun menggunakan Kinesis Data 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 sisa bagian ini, 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 Kinesis Data Firehose dengan penyimpanan Amazon S3. Anda juga dapat mengatur aliran pengiriman Kinesis Data Firehose dengan pengaturan yang berbeda. Untuk informasi selengkapnya, lihat [Membuat Aliran Pengiriman Kinesis Data Firehose](#).

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 Kinesis Data Firehose untuk akun yang sama dan aliran pengiriman lintas wilayah didukung.

Topik

- [Langkah 1: Buat aliran pengiriman Kinesis Data Firehose](#)
- [Langkah 2: Buat tujuan](#)
- [Langkah 3: Tambah/validasi izin IAM untuk tujuan lintas akun](#)
- [Langkah 4: Buat filter berlangganan](#)
- [Memvalidasi alur peristiwa log](#)
- [Memodifikasi keanggotaan tujuan saat runtime](#)

Langkah 1: Buat aliran pengiriman Kinesis Data Firehose

⚠ Important

Sebelum Anda menyelesaikan langkah-langkah berikut, Anda harus menggunakan kebijakan akses, sehingga Kinesis Data Firehose dapat mengakses bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat [Mengontrol Akses](#) di Panduan Pengembang Amazon Kinesis 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 Kinesis Data Firehose untuk 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 IAM role yang memberikan izin kepada Kinesis Data 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 Kinesis Data 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 Kinesis Data 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 pengaliran Kinesis Data Firehose yang Anda buat di [Langkah 1: Buat aliran pengiriman Kinesis Data 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`. `DeliveryStreamArn` ARN, karena Anda harus menggunakaninya di langkah selanjutnya. Contoh output dari perintah ini:

```

        "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 dalam aliran Firehose Data Kinesis Anda. Pertama, Anda harus membuat kebijakan kepercayaan dalam file `~/TrustPolicyForCWL.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 ~/PermissionsForCWL.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 Kinesis Data Firehose dalam status aktif dan Anda telah membuat peran IAM, Anda dapat CloudWatch membuat tujuan Log.
- 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 (1111111111111) 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 PutSubscriptionFilter tindakan 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 Kinesis Data 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 ke [Langkah 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, akun123456789012:

```
{  
    "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, akun123456789012:

```
{  
    "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 11111111111 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 kredensial "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.

Ketika Anda memasukkan perintah berikut, pastikan Anda masuk sebagai pengguna IAM atau menggunakan peran IAM yang Anda tambahkan kebijakan untuk, in. [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 Kinesis Data 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 Kinesis. Data mulai muncul di bucket Amazon S3 Anda berdasarkan interval buffer waktu yang ditetapkan pada aliran pengiriman Kinesis Data 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": "2021/02/08/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 PutDestinationPolicy 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"

{
    "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        {\n            \"Sid\" : \"\",\\n            \"Effect\" : \"Allow\",\\n            \"Principal\" : {\n                \"AWS\" : \"111111111111 \"\\n            },\\n            \"Action\n\" : \"logs:PutSubscriptionFilter\",\\n            \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\\n        }\\n    ]\\n}\\\n",
        "arn": "arn:aws:logs:us-east-1:
222222222222:destination:testFirehoseDestination",
        "creationTime": 1612256124430
    }
}
}

```

- 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"
        }
    ]
}
```

- 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 pemanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggil 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 global aws :SourceArn atau global dalam kebijakan sumber daya untuk membatasi cakupan izin yang Anda berikan kepada CloudWatch Log untuk menulis data ke Kinesis Data Streams dan Kinesis Data Firehose.

Nilai aws :SourceArn harus membatasi izin hanya untuk akun yang menulis dan menerima data.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global aws :SourceArn dengan ARN sumber daya penuh. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks SourceArn global aws: dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, arn:aws:servicename::123456789012: *.

Kebijakan yang didokumentasikan untuk memberikan akses ke CloudWatch Log untuk menulis data ke Kinesis Data Streams dan Kinesis Data [Langkah 1: Buat tujuan](#) Firehose [Langkah 2: Buat tujuan](#) di dan menunjukkan bagaimana Anda dapat menggunakan SourceArn aws: global condition context key untuk membantu mencegah masalah deputi yang membingungkan.

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, lihat [CloudWatch 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](#)
- [Menggunakan pola filter untuk mencocokkan istilah dengan ekspresi reguler \(regex\)](#)
- [Menggunakan pola filter untuk mencocokkan istilah dalam peristiwa log tidak terstruktur](#)
- [Menggunakan pola filter untuk mencocokkan istilah dalam peristiwa log JSON](#)
- [Memudahkan log acara yang dipisahkan dengan spasi](#)

Sintaks ekspresi reguler (regex) yang didukung

Sintaks ekspresi filter

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 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 dibatasi 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 jangka waktu proses. 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 jangka waktu proses. Misalnya, %ab*c% dapat mencocokkan “ac”, “abc”, dan “abbcc”; %ab[0-9]*% dapat mencocokkan “ab”, “ab0”, dan “ab129”.
- +: Cocokkan satu atau lebih contoh dari jangka waktu persidangan. Misalnya, %ab+c% dapat mencocokkan “abc”, “abbc”, dan “abbcc”, tetapi tidak “ac”.
- .: Cocokkan dengan 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. hh menentukan 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% pertandaan:; dan %\x28% pertandaan(.

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 “ERROR” cocok dengan pesan log acara yang berisi istilah ini, seperti berikut:

- [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

Tanda peka huruf besar atau 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 “ERROR” cocok dengan pesan log acara yang berisi istilah ini, seperti berikut:

- [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
```

Pola filter “ERROR” cocok dengan pesan log acara yang berisi istilah ini, seperti berikut:

- [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. 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 “ERROR” cocok dengan pesan log acara yang berisi istilah ini, seperti berikut:

- [ERROR 400] BAD REQUEST
- [ERROR 401] UNAUTHORIZED REQUEST
- [ERROR 419] MISSING ARGUMENTS
- [ERROR 420] INVALID ARGUMENTS

 Note

Tanda tanya (“?”) dengan pola filter lainnya, seperti menyertakan dan mengecualikan istilah. Jika Anda menggabungkan “?” Tanda tanya (“?”) Tanda tanya.

Misalnya, pola filter berikut cocok dengan semua peristiwa yang mengandung kata REQUEST, tetapi tanda tanya (“?”) Tanda filter tidak berpengaruh.

```
?ERROR ?ARGUMENTS REQUEST
```

Log acara log 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 “ERROR” cocok dengan pesan log acara yang berisi istilah ini, seperti berikut:

- [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 menunjuk ke array atau objek, filter tidak akan diterapkan karena format log tidak cocok dengan filter. 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

Tanda tanya (“?”). 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 menunjuk ke array atau objek, filter tidak akan diterapkan karena format log tidak cocok dengan filter. 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 (“! Tanda tanya (“?”)

- 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. ISVariabel dapat mencocokkan bidang yang berisi nilaiNULL, TRUE, atauFALSE. Pola filter berikut mengembalikan log JSON di mana nilai SomeObject adalahNULL.

```
{ $.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" }
```

Memudahkan log acara yang dipisahkan dengan spasi

Memudahkan log acara yang dipisahkan dengan spasi

Memudahkan log acara yang dipisahkan dengan spasi 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 elipsis (...) 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]
```

Memudahkan log acara yang dipisahkan dengan spasi

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]
```

Mengaktifkan logging dari layanan AWS

Meskipun banyak layanan mempublikasikan log hanya ke CloudWatch Log, beberapa AWS layanan dapat mempublikasikan log langsung ke Amazon Simple Storage Service atau Amazon Kinesis Data Firehose. Jika persyaratan utama Anda untuk log adalah penyimpanan atau pemrosesan di salah satu layanan ini, Anda dapat dengan mudah membuat layanan yang menghasilkan log mengirimkannya langsung ke Amazon S3 atau Kinesis Data Firehose tanpa pengaturan tambahan.

Meskipun log dipublikasikan langsung ke Amazon S3 atau Kinesis Data Firehose, ada biaya yang dikenakan. 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 Didukung [Izin V1] dan Didukung [Izin V2] dalam tabel. Untuk informasi tentang izin yang diperlukan ini, lihat bagian setelah tabel.

Jenis log	CloudWatch Logs	Amazon S3	Kinesis Data Firehose
Log akses Amazon API Gateway	Didukung [Izin V1]		
AWS AppSync log	Didukung		
Log MySQL Amazon Aurora	Didukung		
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	Kinesis Data 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 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		
AWS Fargate log	Didukung		
AWS Fault Injection Service log percobaan		Didukung [Izin V1]	
Amazon FinSpace	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
AWS Global Accelerator log aliran		Didukung [Izin V1]	

Jenis log	CloudWatch Logs	Amazon S3	Kinesis Data Firehose
AWS Glue log pekerjaan	Didukung		
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		
AWS Log Firewall Jaringan	Didukung [Izin V1]	Didukung [Izin V1]	Didukung [Izin V1]
Log akses Network Load Balancer		Didukung [Izin V1]	
OpenSearch log	Didukung		

Jenis log	CloudWatch Logs	Amazon S3	Kinesis Data Firehose
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		
File umpan data Instans Spot EC2		Didukung [Izin V1]	
AWS Step Functions Alur Kerja Ekspres dan Log Alur Kerja Standar	Didukung [Izin V1]		

Jenis log	CloudWatch Logs	Amazon S3	Kinesis Data Firehose
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 [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 CodeWhisperer	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 Kinesis Data 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.

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

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 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

Important

Saat Anda menyiapkan jenis log dalam daftar berikut untuk dikirim ke Amazon S3, AWS membuat atau mengubah kebijakan sumber daya yang terkait dengan bucket S3 yang menerima log, jika diperlukan. Lanjutkan membaca bagian ini untuk melihat detailnya.

Bagian ini berlaku ketika jenis log berikut dikirim ke Amazon S3:

- CloudFront log akses dan log akses streaming. CloudFront menggunakan model izin yang berbeda dari layanan lain dalam daftar ini. Untuk informasi selengkapnya, lihat [Izin yang diperlukan untuk mengonfigurasi pencatatan log standar dan untuk mengakses berkas log Anda](#).
- Umpan data Instans Spot Amazon EC2
- AWS Global Accelerator log aliran
- Log broker Amazon Managed Streaming for Apache Kafka
- Log akses Penyeimbang Beban Jaringan
- AWS Log Firewall Jaringan
- log alur Amazon Virtual Private Cloud

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.

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

baruStatement, 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 dikirim 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 yang dikirim ke Kinesis Data Firehose

Bagian ini berlaku ketika jenis log yang tercantum dalam tabel di bagian sebelumnya dikirim ke Kinesis Data Firehose:

Izin pengguna

Untuk dapat mengatur pengiriman salah satu jenis log ini ke Kinesis Data Firehose untuk pertama kalinya, Anda harus masuk ke akun dengan izin berikut.

- `logs>CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam>CreateServiceLinkedRole`

Jika salah satu jenis log ini sudah dikirim ke Kinesis Data Firehose, maka untuk mengatur pengiriman dari salah satu jenis log ini ke Kinesis Data Firehose Anda hanya perlu memiliki izin `logs>CreateLogDelivery` dan `firehose:TagDeliveryStream`.

Peran IAM yang digunakan untuk izin

Karena Kinesis Data Firehose tidak menggunakan AWS kebijakan sumber daya, menggunakan peran IAM saat menyiapkan log ini untuk dikirim ke Kinesis Data 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 Kinesis Data 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"
}
]
}
```

Pencatatan 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 Kinesis Data Firehose.

Untuk mengonfigurasi pengiriman log antara AWS layanan yang didukung dan tujuan, Anda harus melakukan hal berikut:

- Buat sumber pengiriman, yang merupakan objek logis yang mewakili sumber daya yang sebenarnya mengirim log. Untuk informasi lebih lanjut, lihat [PutDeliverySource](#).
- Buat tujuan pengiriman, yang merupakan objek logis yang mewakili tujuan pengiriman yang sebenarnya. Untuk informasi lebih lanjut, lihat [PutDeliveryDestination](#).
- Jika Anda mengirimkan log lintas akun, Anda harus menggunakan [PutDeliveryDestinationPolicy](#) di akun tujuan untuk menetapkan IAM kebijakan ke tujuan. Kebijakan ini memungkinkan pengiriman ke tujuan tersebut.
- Gunakan [CreateDelivery](#) untuk membuat pengiriman dengan memasangkan tepat satu sumber pengiriman dan satu tujuan pengiriman.

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.

Selain izin yang tercantum di bagian berikut, jika Anda menyiapkan pengiriman log menggunakan konsol alih-alih API, Anda juga memerlukan izin tambahan berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLogDeliveryActions",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ]
    }
  ]
}
```

```
        "logs:DescribeLogGroups",
        "firehose>ListDeliveryStreams",
        "firehose:DescribeDeliveryStream",
        "s3>ListAllMyBuckets",
        "s3>ListBucket",
        "s3:GetBucketLocation"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:log-group:*",
        "arn:aws:firehose:region:account-id:deliverystream/*",
        "arn:aws:s3:::/*"
    ]
}
]
```

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": "AllowLogDeliveryActions",
        "Effect": "Allow",
        "Action": [
            "logs:PutDeliverySource",
            "logs:GetDeliverySource",
            "logs>DeleteDeliverySource",
            "logs:DescribeDeliverySources",
            "logs:PutDeliveryDestination",
            "logs:GetDeliveryDestination",
            "logs>DeleteDeliveryDestination",
            "logs:DescribeDeliveryDestinations",
            "logs>CreateDelivery",
            "logs:GetDelivery",
            "logs>DeleteDelivery",
            "logs:DescribeDeliveries",
            "logs:PutDeliveryDestinationPolicy",
            "logs:GetDeliveryDestinationPolicy",
            "logs>DeleteDeliveryDestinationPolicy"
        ]
    }]
}
```

```

    ],
    "Resource": [
        "arn:aws:logs:region:account-id:delivery-source:*",
        "arn:aws:logs:region:account-id:delivery:*",
        "arn:aws:logs:region:account-id:delivery-destination:*
    ]
},
{
    "Sid": "AllowUpdatesToResourcePolicyCWL",
    "Effect": "Allow",
    "Action": [
        "logs:PutResourcePolicy",
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:log-group:*
    ]
}
]
}

```

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": "AllowLogDeliveryActions",
    "Effect": "Allow",
    "Action": [
      "logs:PutDeliverySource",
      "logs:GetDeliverySource",
      "logs>DeleteDeliverySource",
    ]
  }]
}
```

```

        "logs:DescribeDeliverySources",
        "logs:PutDeliveryDestination",
        "logs:GetDeliveryDestination",
        "logs>DeleteDeliveryDestination",
        "logs:DescribeDeliveryDestinations",
        "logs>CreateDelivery",
        "logs:GetDelivery",
        "logs>DeleteDelivery",
        "logs:DescribeDeliveries",
        "logs:PutDeliveryDestinationPolicy",
        "logs:GetDeliveryDestinationPolicy",
        "logs>DeleteDeliveryDestinationPolicy"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:delivery-source:*",
        "arn:aws:logs:region:account-id:delivery:*",
        "arn:aws:logs:region:account-id:delivery-destination:*
    ]
},
{
    "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 baruStatement, 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 dikirim 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 yang dikirim ke Kinesis Data Firehose

Izin pengguna

Untuk mengaktifkan pengiriman log ke Kinesis Data Firehose, Anda harus masuk dengan izin berikut.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    "Sid": "AllowLogDeliveryActions",
    "Effect": "Allow",
    "Action": [
        "logs:PutDeliverySource",
        "logs:GetDeliverySource",
        "logs:DeleteDeliverySource",
        "logs:DescribeDeliverySources",
        "logs:PutDeliveryDestination",
        "logs:GetDeliveryDestination",
        "logs:DeleteDeliveryDestination",
        "logs:DescribeDeliveryDestinations",
        "logs>CreateDelivery",
        "logs:GetDelivery",
        "logs:DeleteDelivery",
        "logs:DescribeDeliveries",
        "logs:PutDeliveryDestinationPolicy",
        "logs:GetDeliveryDestinationPolicy",
        "logs:DeleteDeliveryDestinationPolicy"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:delivery-source:*",
        "arn:aws:logs:region:account-id:delivery:*",
        "arn:aws:logs:region:account-id:delivery-destination:*
    ]
},
{
"Version": "2012-10-17",
"Statement": [
    {
        "Sid": "AllowUpdatesToResourcePolicyFH",
        "Effect": "Allow",
        "Action": [
            "firehose:TagDeliveryStream",
            "iam>CreateServiceLinkedRole"
        ],
        "Resource": [
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"
        ]
    }
]
}]
```

```
}
```

Peran IAM yang digunakan untuk izin sumber daya

Karena Kinesis Data Firehose tidak menggunakan AWS kebijakan sumber daya, menggunakan peran IAM saat menyiapkan log ini untuk dikirim ke Kinesis Data 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 Kinesis Data 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"
}
]
```

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 pemanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggil dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global dalam kebijakan sumber daya untuk membatasi izin yang diberikan CloudWatch Log dan Amazon S3 ke layanan yang menghasilkan log. Jika Anda menggunakan kedua kunci konteks kondisi global, `aws:SourceAccount` nilai dan akun dalam `aws:SourceArn` nilai harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Nilai `aws:SourceArn` harus berupa ARN dari sumber pengiriman yang menghasilkan log.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN sumber daya penuh. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui.

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
<u>AWSServiceRoleForLogDelivery kebijakan peran terkait layanan — Pembaruan ke kebijakan</u> yang ada	CloudWatch Log mengubah izin dalam kebijakan IAM yang terkait dengan peran terkait AWSServiceRoleForLogDelivery layanan. Perubahan berikut dibuat: <ul style="list-style-type: none">• Kunci firehose: ResourceTag/LogDeliveryEnabled": "true" kondisi diubah menjadiaws: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

Eksport 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 empat di akun yang sama atau akun lain.

Anda dapat melakukan hal 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

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.

A blue circular icon containing a white lowercase letter 'i', representing a note or informational message.

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&/\1/' | sort -z
```

Perintah utilitas berikut mengurutkan file.gz dari beberapa subfolder.

```
find ./*/ -type f -exec zcat {} + | sed -r 's/^[\0-9]+/\x08/' | 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](#)
- [Ekspor data log ke Amazon S3 menggunakan konsol](#)
- [Ekspor data log ke Amazon S3 menggunakan AWS CLI](#)
- [Jelaskan tugas ekspor](#)
- [Membatalkan tugas ekspor](#)

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 log acara dalam grup log yang diserap 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](#)
- [Ekspor lintas akun](#)

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.](#)
- [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 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, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti petunjuk dalam [Buat set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti petunjuk dalam [Membuat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) di Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti petunjuk dalam [Membuat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk di [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 dieksport 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 akan menjadi akun dari mana 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` ke 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 keysEnkripsi 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 didokumentasikan[Langkah 2: Siapkan izin akses](#).
2. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
3. Di 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.](#)
- [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 eksport](#)

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 eksport 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:kms:region:account-id:key-id/*"  
        }  
    ]  
}
```

```
{  
    "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 Berikutnya.
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, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti petunjuk dalam [Buat set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti petunjuk dalam [Membuat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) di Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti petunjuk dalam [Membuat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk di [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 dieksport 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 akan menjadi akun dari mana 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` ke 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 keysEnkripsi 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": "Allow CloudWatch Metrics Put",  
            "Effect": "Allow",  
            "Principal": "logs.Region.amazonaws.com",  
            "Action": "kms:Encrypt",  
            "Resource": "arn:aws:kms:Region:AccountID::/alias/LogDeliveryKey"  
        }  
    ]  
}
```

```
        "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 didokumentasikan [Langkah 2: Siapkan izin akses](#).
2. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
3. Di 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 yang diberi nama 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](#)
- [Ekspor lintas akun](#)

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](#)
- [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

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, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti petunjuk dalam [Buat set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti petunjuk dalam [Membuat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) di Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti petunjuk dalam [Membuat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk di [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 akan menjadi akun dari mana pengguna dapat mengekspor data log ke bucket S3 Anda. `aws:SourceArnKuncinya` 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",  
                        "AccountId3"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        "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 Anda dengan menggunakan [put-bucket-policy](#) perintah. 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 keysEnkripsi 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 didokumentasikan [Langkah 2: Siapkan izin akses](#).
2. Pada prompt perintah, gunakan [create-export-task](#) perintah 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 144149400000 --destination "my-exported-logs" --destination-prefix "export-task-output"
```

Berikut ini adalah output contoh.

```
{  
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"  
}
```

Ekspor lintas akun

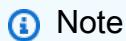
Jika bucket Amazon S3 berada di akun yang berbeda dari log yang sedang diekspor, gunakan petunjuk di bagian ini.

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, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti petunjuk dalam [Buat set izin](#) dalam Panduan Pengguna AWS IAM Identity Center.

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti petunjuk dalam [Membuat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) di Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti petunjuk dalam [Membuat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk di [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 akan menjadi akun dari mana pengguna dapat mengekspor data log ke bucket S3 Anda. `aws:SourceArnKuncinya` 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",  
                        "AccountId3"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        "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 Anda dengan menggunakan [put-bucket-policy](#) perintah. 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 keysEnkripsi 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 didokumentasikan [Langkah 2: Siapkan izin akses](#).
2. Pada prompt perintah, gunakan [create-export-task](#) perintah 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-tasks](#) perintah 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-task](#) perintah berikut:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-
aade86253e66"
```

Anda dapat menggunakan [describe-export-tasks](#) perintah untuk memverifikasi bahwa tugas telah dibatalkan dengan sukses.

Streaming CloudWatch Log data ke Amazon OpenSearch Service

Anda dapat mengonfigurasi grup CloudWatch log Logs untuk melakukan pengaliran data yang diterima ke klaster Amazon OpenSearch Service dalam hampir waktu nyata melalui langganan CloudWatch Logs. Untuk informasi selengkapnya, lihat [Pemrosesan data log secara real-time dengan langganan](#).

Tergantung pada jumlah data log yang dialirkkan, Anda mungkin ingin menetapkan batas eksekusi serentak tingkat fungsi pada fungsi. Untuk informasi lebih lanjut, lihat [penskalaan fungsi Lambda](#).

 Note

Streaming data CloudWatch Log dalam jumlah besar ke OpenSearch Layanan dapat mengakibatkan biaya penggunaan yang tinggi. Kami merekomendasikan agar Anda membuat Anggaran diAWS Billing and Cost Management konsol. Untuk informasi lebih lanjut, lihat [Mengelola biaya Anda dengan AWS Anggaran](#).

Prasyarat

Sebelum memulai, buat domain OpenSearch Layanan. Domain dapat memiliki akses publik atau akses VPC, tetapi kemudian Anda tidak dapat memodifikasi jenis akses setelah domain dibuat. Anda mungkin ingin meninjau pengaturan domain OpenSearch Service Anda nanti, dan memodifikasi konfigurasi klaster Anda berdasarkan jumlah data klaster yang akan diproses. 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. Di panel navigasi, pilih Grup log.
3. 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 ARN domain dan titik akhir.
6. Untuk Lambda IAM Execution Role (Peran Eksekusi Lambda IAM), pilih IAM role yang harus digunakan Lambda ketika 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 `AWSLambdaVPCAccessExecutionRole` kebijakan yang dilampirkan. Kebijakan yang dikelola Amazon ini memberikan akses Lambda ke VPC pelanggan, memungkinkan Lambda untuk menulis ke OpenSearch titik akhir di VPC.

7. Untuk Log Format (Format Log), pilih format log.

8. Untuk Subscription filter Pattern (Pola filter Langganan), ketik istilah atau pola yang akan dicari di log acara Anda. Hal ini memastikan bahwa Anda hanya mengirim data yang penting bagi Anda ke OpenSearch klaster. Untuk informasi selengkapnya, lihat [Membuat metrik dari peristiwa log menggunakan filter](#).
9. (Opsional) Untuk Select log data to Test (Pilih data log), pilih pengaliran log, lalu pilih Test Pattern (Uji Pola) untuk memverifikasi bahwa filter pencarian Anda memberikan hasil yang diharapkan.
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 merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Contoh lintas layanan adalah contoh aplikasi yang bekerja di beberapa Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Contoh kode

- [Tindakan untuk CloudWatch Log menggunakan AWS SDK](#)
 - [Mengaitkan AWS KMS kunci dengan grup CloudWatch log Log menggunakan AWS SDK](#)
 - [Membatalkan tugas ekspor CloudWatch Log menggunakan AWS SDK](#)
 - [Membuat grup CloudWatch log Log menggunakan AWS SDK](#)
 - [Membuat aliran CloudWatch log Log menggunakan AWS SDK](#)
 - [Membuat filter langganan CloudWatch Log menggunakan AWS SDK](#)
 - [Membuat tugas ekspor CloudWatch Log menggunakan AWS SDK](#)
 - [Menghapus grup CloudWatch log Log menggunakan AWS SDK](#)
 - [Menghapus filter langganan CloudWatch Log menggunakan AWS SDK](#)
 - [Jelaskan filter langganan CloudWatch Log menggunakan AWS SDK](#)
 - [Menjelaskan tugas ekspor CloudWatch Log menggunakan AWS SDK](#)
 - [Jelaskan grup CloudWatch log log menggunakan AWS SDK](#)
- [Contoh lintas layanan untuk CloudWatch Log menggunakan AWS SDK](#)
- [Menggunakan peristiwa terjadwal untuk menginvokasi fungsi Lambda](#)

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-contoh

- [Mengaitkan AWS KMS kunci dengan grup CloudWatch log Log menggunakan AWS SDK](#)
- [Membatalkan tugas ekspor CloudWatch Log menggunakan AWS SDK](#)
- [Membuat grup CloudWatch log Log menggunakan AWS SDK](#)
- [Membuat aliran CloudWatch log Log menggunakan AWS SDK](#)
- [Membuat filter langganan CloudWatch Log menggunakan AWS SDK](#)
- [Membuat tugas ekspor CloudWatch Log menggunakan AWS SDK](#)
- [Menghapus grup CloudWatch log Log menggunakan AWS SDK](#)
- [Menghapus filter langganan CloudWatch Log menggunakan AWS SDK](#)
- [Jelaskan filter langganan CloudWatch Log menggunakan AWS SDK](#)
- [Menjelaskan tugas ekspor CloudWatch Log menggunakan AWS SDK](#)
- [Jelaskan grup CloudWatch log log menggunakan AWS SDK](#)

Mengaitkan AWS KMS kunci dengan grup CloudWatch log Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara mengaitkan AWS KMS kunci dengan grup CloudWatch log Log yang ada.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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. The example was created using the
/// AWS SDK for .NET version 3.7 and .NET Core 5.0.
/// </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/7c9ecc2-38cb-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 [AssociateKmsKey](#) di Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membatalkan tugas ekspor CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membatalkan tugas ekspor CloudWatch Log yang ada.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

```
/// <summary>
/// Shows how to cancel an Amazon CloudWatch Logs export task. The example
/// uses the AWS SDK for .NET version 3.7 and .NET Core 5.0.
/// </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 [CancelExportTask](#) di Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membuat grup CloudWatch log Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat grup CloudWatch log Log baru.

.NET

AWS SDK for .NET

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to create an Amazon CloudWatch Logs log group. The example
/// was created using the AWS SDK for .NET version 3.7 and .NET Core 5.0.
/// </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 [CreateLogGroup](#) di Referensi AWS SDK for .NET API.

JavaScript

SDK untuk JavaScript (v3)

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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 [CreateLogGroup](#) di Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membuat aliran CloudWatch log Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat aliran CloudWatch log Log baru.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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. The example was created using the AWS SDK for .NET version
/// 3.7 and .NET Core 5.0.
/// </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 [CreateLogStream](#) di Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membuat filter langganan CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat filter langganan Amazon CloudWatch Logs.

C++

SDK for C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di [Repositori Contoh KodeAWS](#).

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 berlangganan.

```
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 [PutSubscriptionFilter](#) di Referensi AWS SDK for C++ API.

Java

SDK for Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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 [PutSubscriptionFilter](#) di 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 [Repositori Contoh KodeAWS](#).

```
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 [PutSubscriptionFilter](#) di Referensi AWS SDK for JavaScript API.
- SDK untuk JavaScript (v2)

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur 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 [PutSubscriptionFilter](#) di Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membuat tugas ekspor CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat tugas ekspor CloudWatch Log baru.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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. The example was created with the AWS SDK for .NET version 3.7 and
/// .NET Core 5.0.
/// </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 taskId = "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 [CreateExportTask](#) di Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Menghapus grup CloudWatch log Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menghapus grup CloudWatch log Log yang ada.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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. The example was created using the
/// AWS SDK for .NET version 3.7 and .NET Core 5.0.
/// </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 [DeleteLogGroup](#) di Referensi AWS SDK for .NET API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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 [DeleteLogGroup](#) di Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Menghapus filter langganan CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menghapus filter langganan Amazon CloudWatch Logs.

C++

SDK for C++

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di [Repositori Contoh KodeAWS](#).

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 [DeleteSubscriptionFilter](#) di Referensi AWS SDK for C++ API.

Java

SDK for Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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 [DeleteSubscriptionFilter](#) di 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 [Repositori Contoh KodeAWS](#).

```
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 [DeleteSubscriptionFilter](#) di Referensi AWS SDK for JavaScript API.

SDK untuk JavaScript (v2)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
// 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 lengkapnya, silakan lihat [Panduan Developer AWS SDK for JavaScript](#).
- Untuk detail API, lihat [DeleteSubscriptionFilter](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK for Kotlin

Note

Ini adalah dokumentasi prarilis untuk fitur dalam rilis pratinjau. Dokumentasi dapat berubah.

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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 [DeleteSubscriptionFilter](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan filter langganan CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara mendeskripsikan filter langganan Amazon CloudWatch Logs yang ada.

C++

SDK for C++

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di [Repositori Contoh KodeAWS](#).

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 [DescribeSubscriptionFilters](#) di Referensi AWS SDK for C++ API.

Java

SDK for Java 2.x

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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 [DescribeSubscriptionFilters](#) di 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 [Repositori Contoh Kode AWS](#).

```
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 [DescribeSubscriptionFilters](#) di Referensi AWS SDK for JavaScript API. SDK untuk JavaScript (v2)

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur 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 DeveloperAWS SDK for JavaScript](#).
- Untuk detail API, lihat [DescribeSubscriptionFilters](#) di ReferensiAWS SDK for JavaScript API.

Kotlin

SDK for Kotlin

Note

Ini adalah dokumentasi prarilis untuk fitur dalam rilis pratinjau. Dokumentasi dapat berubah.

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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 [DescribeSubscriptionFilters](#) di AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Menjelaskan tugas ekspor CloudWatch Log menggunakan AWS SDK

Contoh kode berikut menunjukkan bagaimana mendeskripsikan tugas ekspor CloudWatch Log.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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. The example was created using the AWS SDK for .NET  
/// version 3.7 and .NET Core 5.0.  
/// </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 [DescribeExportTasks](#) di Referensi AWS SDK for .NET API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan grup CloudWatch log menggunakan AWS SDK

Contoh kode berikut menunjukkan cara mendeskripsikan grup CloudWatch log Log.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh KodeAWS](#).

```
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. The example was created
/// using the AWS SDK for .NET version 3.7 and .NET Core 5.0.
/// </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 [DescribeLogGroups](#) di Referensi AWS SDK for .NET API.

JavaScript

SDK untuk JavaScript (v3)

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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 [DescribeLogGroups](#) di Referensi AWS SDK for JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-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-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.

- Menyebarluaskan 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, lihat [Menggunakan CloudWatch Log dengan AWS SDK](#). Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Keamanan di AmazonCloudWatchBeberapa catatan

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

Keamanan adalah tanggung jawab bersama antara 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 layanan AWS di Cloud AWS. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi efektivitas keamanan kami secara berkala sebagai bagian dari [Program Kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku untuk WorkSpaces, Lihat [AWS Layanan dalam Lingkup oleh Program Kepatuhan](#).
- Keamanan dalam cloud – Tanggung jawab Anda ditentukan oleh layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, mencakup kepekaan data Anda, persyaratan perusahaan, serta peraturan perundungan yang berlaku

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AmazonCloudWatchLog. Ini menunjukkan kepada Anda cara mengonfigurasi AmazonCloudWatchLog untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan lainnya AWS layanan yang membantu Anda memantau dan mengamankan CloudWatchLog sumber daya.

Konten

- [Perlindungan Data di Amazon CloudWatch Logs](#)
- [Manajemen identitas dan akses untuk Amazon CloudWatch Logs](#)
- [Validasi kepatuhan untuk Amazon Logs CloudWatch](#)
- [Ketahanan di Amazon CloudWatch Logs](#)
- [Keamanan infrastruktur di Amazon CloudWatch Logs](#)
- [Menggunakan CloudWatch Log dengan titik akhir VPC antarmuka](#)

Perlindungan Data di Amazon CloudWatch Logs

Note

Selain informasi berikut tentang perlindungan data umum diAWS, CloudWatch Log juga memungkinkan Anda untuk melindungi data sensitif dalam peristiwa log dengan menutupi itu. Untuk informasi selengkapnya, lihat [Membantu melindungi data log sensitif dengan masking](#).

[Model tanggung jawabAWS bersama model](#) diterapkan untuk perlindungan data di Amazon CloudWatch Logs. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk memberikan perlindungan terhadap infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Konten ini meliputi konfigurasi keamanan dan tugas-tugas pengelolaan untuk berbagai layanan Layanan AWS yang Anda gunakan. Untuk informasi lebih lanjut tentang privasi data, lihat [FAQ tentang Privasi Data](#). Untuk informasi tentang perlindungan data di Eropa, lihat postingan blog [Model Tanggung Jawab Bersama AWS dan GDPR](#) di Blog Keamanan AWS.

Untuk tujuan perlindungan data, kami merekomendasikan agar Anda melindungiAkun AWS kredensyal dan menyiapkan pengguna individu denganAWS IAM Identity Center atauAWS Identity and Access Management (IAM). Dengan cara tersebut, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tugas pekerjaan mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusiAWS enkripsi, bersama semua kontrol keamanan default di dalamnyaLayanan AWS.
- Gunakan layanan keamanan terkelola lanjutan seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi lebih lanjut tentang titik akhir FIPS yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat menyarankan agar Anda tidak memasukkan informasi rahasia atau sensitif apa pun, seperti alamat email pelanggan Anda, ke dalam tanda atau kolom teks bebas seperti kolom Nama. Hal ini termasuk saat Anda bekerja dengan CloudWatch Logs atau orang lain yang menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau kolom teks bebas yang digunakan untuk nama dapat digunakan untuk penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menyarankan jangan menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi saat tidak aktif

CloudWatch Logs melindungi data at rest menggunakan enkripsi. Semua grup log dienkripsi. Secara default, layanan CloudWatch Logs mengelola kunci enkripsi sisi server.

Jika Anda ingin mengelola kunci yang digunakan untuk mengenkripsi dan mendekripsi log Anda, gunakan kunci utama pelanggan (CMK) dari AWS Key Management Service. Untuk informasi selengkapnya, lihat [Enkripsi data log di CloudWatch Log menggunakan AWS Key Management Service](#).

Enkripsi dalam transit

CloudWatch Logs menggunakan end-to-end enkripsi data saat transit. Layanan CloudWatch Logs mengelola kunci enkripsi sisi server.

Manajemen identitas dan akses untuk Amazon CloudWatch Logs

Akses ke Amazon CloudWatch Log memerlukan kredensial yang AWS dapat digunakan untuk mengautentikasi permintaan Anda. Kredensial 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, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti petunjuk dalam [Buat set izin](#) dalam Panduan Pengguna AWS IAM Identity Center.

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti petunjuk dalam [Membuat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) di Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti petunjuk dalam [Membuat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk di [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 disarankan untuk membaca gambaran umum terlebih dahulu.

- [Ikhtisar mengelola izin akses ke sumber daya CloudWatch Log](#)
- [Menggunakan kebijakan berbasis identitas \(kebijakan IAM\) untuk Log CloudWatch](#)
- [CloudWatch Referensi izin log](#)

Ikhtisar mengelola izin akses ke sumber daya CloudWatch Log

Untuk memberikan akses, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti petunjuk dalam [Buat set izin](#) dalam Panduan Pengguna AWS IAM Identity Center.

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti petunjuk dalam [Membuat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) di Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti petunjuk dalam [Membuat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk di [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 <code>describe-log-groups</code> CLI dan API <code>DescribeLogGroups</code> <code>arn:aws:logs:<i>region:account-id</i>:log-group:<i>log_group_name</i></code>

Jenis sumber daya	Format ARN
	<i>arn:aws:logs: wilayah: account-id:log-group: log_group_name : *</i>
Pengaliran log	<i>arn:aws:logs: wilayah: account-id:log-group: log_group_name:log-stream: log-stream-name</i>
Tujuan	<i>arn:aws:logs:region:account-id :destination:destination_name</i>

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. Untuk contoh kebijakan yang mencakup CloudWatch Log, lihat [Menggunakan 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 kredensial 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, yang menjadi milik peran tersebut, 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 [Berbagi data log lintas akun dengan langganan](#).

Topik

- [Izin grup log dan Wawasan Kontributor](#)
- [Kebijakan berbasis sumber daya](#)

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 [PutDestinationAPI](#). 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 IAMAWS](#) dalam Panduan Pengguna IAM.

Untuk tabel yang menampilkan semua tindakan API CloudWatch Log dan sumber daya yang diterapkan, lihat [CloudWatch 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](#).

Topik ini mencakup hal-hal berikut:

- [Izin 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, lihat [CloudWatch Referensi izin log](#).

Izin 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 dalam [AWS 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. Isinya sebagai berikut:

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

CloudWatchLogsReadOnlyAccess

CloudWatchLogsReadOnlyAccessKebijakan ini memberikan akses hanya-baca ke Log. CloudWatch 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"
    ],
    "Resource": "*"
}
]
```

CloudWatchLogsCrossAccountSharingConfiguration

CloudWatchLogsCrossAccountSharingConfiguration Kebijakan 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
CloudWatchLogsReadOnlyAccess — Permbaruan ke kebijakan yang sudah ada	<p>CloudWatch Log menambahkan izin ke CloudWatchLogsReadOnlyAccess.</p> <p>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.</p>	6 Juni 2023
CloudWatchLogsCrossAccountSharingConfiguration – Kebijakan baru	<p>CloudWatch Log menambahkan kebijakan baru untuk memungkinkan Anda mengelola tautan pengamatan</p>	27 November 2022

Perubahan	Deskripsi	Tanggal
	<p>CloudWatch lintas akun yang berbagi grup CloudWatch log Log.</p> <p>Untuk informasi selengkapnya, lihat CloudWatch observability lintas akun</p>	
<u>CloudWatchLogsFullAccess</u> – Pembaruan pada kebijakan yang sudah ada	<p>CloudWatch Log menambahkan izin ke CloudWatchLogsFullAccess.</p> <p>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.</p>	27 November 2022

Perubahan	Deskripsi	Tanggal
<u>CloudWatchLogsReadOnlyAccess</u> – Pembaruan pada kebijakan yang sudah ada	CloudWatch Log menambahkan izin ke CloudWatchLogsReadOnlyAccess. Izin oam>ListSinks dan oam>ListAttachedLogs 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

- [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"  
            ],  
            "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 ditegakkan.

```
{  
    "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 StopQuery dan 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 Operasi API log	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 Operasi API log	Izin yang diperlukan (tindakan API) Diperlukan untuk membuat grup log baru.
<u>CreateLogStream</u>	<code>logs:CreateLogStream</code> Diperlukan untuk membuat aliran log baru dalam grup log.
<u>DeleteDestination</u>	<code>logs:DeleteDestination</code> Diperlukan untuk menghapus tujuan log dan menonaktifkan penyaring berlangganan apa pun.
<u>DeleteLogGroup</u>	<code>logs:DeleteLogGroup</code> Diperlukan untuk menghapus grup log dan semua peristiwa log yang diarsipkan yang terkait.
<u>DeleteLogStream</u>	<code>logs:DeleteLogStream</code> Diperlukan untuk menghapus aliran log dan peristiwa log yang diarsipkan yang terkait.
<u>DeleteMetricFilter</u>	<code>logs:DeleteMetricFilter</code> Diperlukan untuk menghapus penyaring metrik yang terkait dengan grup log.
<u>DeleteQueryDefinition</u>	<code>logs:DeleteQueryDefinition</code> Diperlukan untuk menghapus definisi kueri yang disimpan di Wawasan CloudWatch Log.
<u>DeleteResourcePolicy</u>	<code>logs:DeleteResourcePolicy</code> Diperlukan untuk menghapus kebijakan sumber daya CloudWatch Log.

CloudWatch Operasi API log	Izin yang diperlukan (tindakan API)
<u>DeleteRetentionPolicy</u>	<code>logs:DeleteRetentionPolicy</code> Diperlukan untuk menghapus kebijakan penyimpanan grup log.
<u>DeleteSubscriptionFilter</u>	<code>logs:DeleteSubscriptionFilter</code> Diperlukan untuk menghapus penyaring berlangganan yang terkait dengan grup log.
<u>DescribeDestinations</u>	<code>logs:DescribeDestinations</code> Diperlukan untuk melihat semua destinasi yang terkait dengan akun.
<u>DescribeExportTasks</u>	<code>logs:DescribeExportTasks</code> Diperlukan untuk melihat semua tugas ekspor yang terkait dengan akun.
<u>DescribeLogGroups</u>	<code>logs:DescribeLogGroups</code> Diperlukan untuk melihat semua grup log yang terkait dengan akun.
<u>DescribeLogStreams</u>	<code>logs:DescribeLogStreams</code> Diperlukan untuk melihat semua aliran log yang terkait dengan grup log.
<u>DescribeMetricFilters</u>	<code>logs:DescribeMetricFilters</code> Diperlukan untuk melihat semua metrik yang terkait dengan grup log.
<u>DescribeQueryDefinitions</u>	<code>logs:DescribeQueryDefinitions</code> Diperlukan untuk melihat daftar definisi kueri yang disimpan di Wawasan CloudWatch Log.

CloudWatch Operasi API log	Izin yang diperlukan (tindakan API)
<u>DescribeQueries</u>	logs:DescribeQueries Diperlukan untuk melihat daftar kueri Wawasan CloudWatch Log yang dijadwalkan, dijalankan, atau baru-baru ini dikeluarkan.
<u>DescribeResourcePolicies</u>	logs:DescribeResourcePolicies Diperlukan untuk melihat daftar kebijakan sumber daya CloudWatch Log.
<u>DescribeSubscriptionFilters</u>	logs:DescribeSubscriptionFilters Diperlukan untuk melihat semua penyaring berlangganan yang terkait dengan grup log.
<u>FilterLogEvents</u>	logs:FilterLogEvents Diperlukan untuk mengurutkan peristiwa log berdasarkan pola penyaringan grup log.
<u>GetLogEvents</u>	logs:GetLogEvents Diperlukan untuk mengambil kejadian log dari aliran log.
<u>GetLogGroupFields</u>	logs:GetLogGroupFields Diperlukan untuk mengambil daftar kolom yang disertakan dalam peristiwa log di grup log.
<u>GetLogRecord</u>	logs:GetLogRecord Diperlukan untuk mengambil rincian dari satu peristiwa log.

CloudWatch Operasi API log	Izin yang diperlukan (tindakan API)
<u>GetQueryResults</u>	logs:GetQueryResults Diperlukan untuk mengambil hasil kueri Wawasan CloudWatch Log.
<u>ListTagsLogGroup</u>	logs>ListTagsLogGroup Diperlukan untuk membuat daftar tag yang terkait dengan grup log.
<u>PutDestination</u>	logs:PutDestination Diperlukan untuk membuat atau memperbarui aliran log tujuan (seperti aliran Kinesis).
<u>PutDestinationPolicy</u>	logs:PutDestinationPolicy Diperlukan untuk membuat atau memperbarui kebijakan akses yang terkait dengan tujuan log yang sudah ada.
<u>PutLogEvents</u>	logs:PutLogEvents Diperlukan untuk mengunggah kumpulan peristiwa log ke aliran log.
<u>PutMetricFilter</u>	logs:PutMetricFilter Diperlukan untuk membuat atau memperbarui penyaring metrik dan mengaitkannya dengan grup log.
<u>PutQueryDefinition</u>	logs:PutQueryDefinition Diperlukan untuk menyimpan kueri di Wawasan CloudWatch Log.

CloudWatch Operasi API log	Izin yang diperlukan (tindakan API)
<u>PutResourcePolicy</u>	<code>logs:PutResourcePolicy</code> Diperlukan untuk membuat kebijakan sumber daya CloudWatch Log.
<u>PutRetentionPolicy</u>	<code>logs:PutRetentionPolicy</code> Diperlukan untuk mengatur jumlah hari untuk menyimpan peristiwa (penyimpanan) log dalam grup log.
<u>PutSubscriptionFilter</u>	<code>logs:PutSubscriptionFilter</code> Diperlukan untuk membuat atau memperbarui penyaring berlangganan dan mengaitkannya dengan grup log.
<u>StartQuery</u>	<code>logs:StartQuery</code> Diperlukan untuk memulai kueri Wawasan CloudWatch Log.
<u>StopQuery</u>	<code>logs:StopQuery</code> Diperlukan untuk menghentikan kueri Wawasan CloudWatch Log yang sedang berlangsung.
<u>TagLogGroup</u>	<code>logs:TagLogGroup</code> Perlu menambahkan atau memperbarui tag grup log.
<u>TestMetricFilter</u>	<code>logs:TestMetricFilter</code> 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 dengan sebuah tautan untuk melihat dokumentasi peran terkait 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 Kinesis Data Firehose. Untuk informasi selengkapnya, lihat [Mengaktifkan logging dari layanan AWS](#).

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 di semua pengaliran Kinesis Data Firehose yang memiliki tanda dengan kunci LogDeliveryEnabled dengan nilai True. Tanda ini secara otomatis dilampirkan ke pengaliran Kinesis Data Firehose saat Anda membuat langganan untuk mengirimkan log ke Kinesis Data 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 Kinesis Data Firehose AWS Management Consoledi AWS CLI, the, AWS atau CloudWatch API, Log akan membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran terkait 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 CloudWatch aliran Firehose Data Kinesis, 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, silakan lihat [Mengedit peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Log CloudWatch

Jika tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran tertaut layanan, sebaiknya 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 AWSServiceRoleForLogDelivery peran terkait layanan

- Berhenti mengirim log langsung ke aliran Kinesis Data Firehose.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran AWSServiceRoleForLogDelivery terkait 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 oleh AWS Layanan Program Kepatuhan](#). Untuk informasi umum, lihat [Program Kepatuhan AWS](#).

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan 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 AWS yang 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 Anda](#) dalam 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.

Membuat CloudWatch log panggilan API Amazon LogsAWS 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 merekam panggilan API yang dibuat oleh atau atas nama AWS akun Anda. Panggilan yang direkam mencakup panggilan dari CloudWatch konsol tersebut dan panggilan kode ke operasi API CloudWatch Logs. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan CloudTrail kejadian ke bucket Amazon S3, termasuk kejadian untuk CloudWatch Log. Jika Anda tidak dapat mengkonfigurasi jejak, Anda masih dapat melihat tindakan terbaru di CloudTrail konsol di Riwayat tindakan. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke CloudWatch Logs, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengonfigurasi dan mengaktifkannya, lihat [Panduan AWS CloudTrail Pengguna](#).

Topik

- [CloudWatch Log informasi di CloudTrail](#)
- [Memahami entri file log](#)

CloudWatch Log informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Saat aktivitas peristiwa yang didukung terjadi di CloudWatch Log, aktivitas tersebut dicatat di CloudTrail peristiwa bersama peristiwa AWS layanan lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat Kejadian dengan Riwayat CloudTrail Kejadian](#).

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

- [Ikhtisar untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengonfigurasi NoS untuk Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima Berkas CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima Berkas CloudTrail Log dari Beberapa Akun](#)

CloudWatch Log mendukung pencatatan tindakan berikut sebagai peristiwa dalam berkas 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 dicatat CloudTrail untuk tindakan API CloudWatch Log berikut:

- [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:

- Jika permintaan tersebut dibuat dengan kredensial pengguna root atau IAM.
- Jika permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna federasi.
- Bawa permintaan dibuat oleh layanan AWS lain.

Untuk informasi lain, lihat [Elemen userIdentity CloudTrail](#).

Memahami entri file log

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

Entri berkas log berikut menunjukkan bahwa pengguna memanggil tindakan log berikut menunjukkan bahwa pengguna memanggil CreateExportTasktindakan CloudWatch log berikut.

```
{  
    "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 yang lama tidak digunakan lagi CloudWatch Log agen. Jika Anda menggunakan Instance Metadata Service Versi 2 (IMDSv2), Anda harus menggunakan terpadu baru CloudWatch agen. Bahkan jika Anda tidak menggunakan IMDSv2, kami sangat menyarankan Anda untuk menggunakan terpadu yang lebih baru CloudWatch agen bukan agen log yang lebih tua. Untuk informasi selengkapnya tentang agen terpadu yang lebih baru, lihat [Mengumpulkan metrik dan log dari instans Amazon EC2 dan server on-premise dengan CloudWatch agen](#).

Untuk informasi tentang migrasi dari yang lebih tua CloudWatch Log agen untuk agen terpadu, lihat [Buat CloudWatch file konfigurasi agen dengan wizard](#).

Yang CloudWatch Agen Log menyediakan cara otomatis untuk mengirim data log ke CloudWatch Log dari instans Amazon EC2. Agen meliputi komponen berikut:

- Plug-in keAWS CLl yang mendorong data log ke CloudWatch Log.
- Skrip (daemon) yang memulai proses untuk mendorong data CloudWatch Log.
- Tugas cron yang memastikan bahwa daemon selalu berjalan.

File konfigurasi agen

Yang CloudWatch File konfigurasi agen Log menjelaskan informasi yang diperlukan oleh CloudWatch Log agen. 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

(Opsiional) 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

Jika diatur ke true (default), ini akan mengaktifkan pengodean konten http gzip untuk mengirim muatan terkompresi ke CloudWatch Log. Hal ini mengurangi penggunaan CPU, menurunkan NetworkOut, dan mengurangi latensi put. Untuk menonaktifkan fitur ini, tambahkan `use_gzip_http_content_encoding = salahkepada[umum]`bagian dari CloudWatch Log file konfigurasi agen, dan kemudian restart agen.

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 penambah nol. 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 nol-padded. 01, 02,..., 12

%d: Tanggal dalam bulan sebagai angka desimal nol-padded. 01, 02,..., 31

%H: Jam (24 jam) sebagai angka desimal nol-padded. 00, 01,..., 23

%I: Jam (12 jam) sebagai angka desimal nol-padded. 01, 02,..., 12

%p: Istilah lokal yang setara dengan AM atau PM.

%M: Menit sebagai angka desimal nol-padded. 00, 01,..., 59

%S: Kedua sebagai angka desimal nol-padded. 00, 01,..., 59

%f: Mikrosekon sebagai angka desimal, nol-padded di sebelah kiri. 000000,..., 999999

%z: Pengimbangan UTC dalam bentuk +HHHHHHHHHHHHMM. +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 berkas log yang ingin Anda dorong 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 didekripsi 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 CloudWatch Agen log dengan proxy HTTP

Anda dapat menggunakan CloudWatch Agen log dengan proxy HTTP.

Note

Proxy HTTP didukung di awslogs-agent-setup.py versi 1.3.8 atau yang lebih baru.

Untuk menggunakan CloudWatch Agen log dengan proxy HTTP

1. Lakukan salah satu dari berikut:

a. Untuk instalasi baru CloudWatch Log agen, jalankan perintah berikut:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -0
```

```
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 untuk Instans Linux.

Dalam nilai untuk http-proxy dan https-proxy, Anda menentukan seluruh URL.

- b. Untuk instalasi yang ada CloudWatch Agen log, edit /var/awslogs/etc/proxy.conf, lalu 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
```

Kompartimentalisasi CloudWatch Log file konfigurasi agen

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/var/awslogs/etc/config/direktori. Saat CloudWatch Agen log dimulai, ini mencakup konfigurasi aliran apa pun di 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 Log Agent

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 script setup untuk menginstal CloudWatch Log agen, Anda dapat menggunakan /var/awslogs/bin/awslogs-version.shuntuk memeriksa versi agen yang Anda gunakan. Versi agen dan dependensi utamanya akan dicetak. Jika Anda menggunakan yum untuk menginstal CloudWatch Log agen, Anda dapat menggunakan "Info yum awslogs" dan "Info yum aws-cli-plugin-cloudwatch-log" untuk memeriksa versi CloudWatch Log agen dan plugin.

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

Yang CloudWatch Agen log menulis peringatan untuk log-nya ketika 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. Yang CloudWatch Agen log dapat dimulai dari tempatnya dihentikan dan melanjutkan 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)?

Yang CloudWatch Log agen
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:  
                    <region>:  
                    <account>/log-group/  
                    <log-group-name>:  
                    <log-stream-name>  
            ]  
        }  
    ]  
}
```

```
    "arn:aws:logs:*:*:*"
]
}
]
}
```

Saya tidak ingin CloudWatch Agen log untuk membuat grup log atau pengaliran 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`.

Memantau dengan CloudWatch metrik

CloudWatch Logs mengirimkan metrik ke Amazon CloudWatch setiap menitnya.

CloudWatch Metrik log

Namespace AWS/Logs mencakup metrik berikut.

Metrik	Deskripsi
CallCount	<p>Jumlah operasi API tertentu yang dilakukan di akun.</p> <p>CallCount adalah metrik penggunaan layanan CloudWatch Log. Untuk informasi selengkapnya, lihat CloudWatch Log metrik penggunaan layanan.</p> <p>Dimensi yang Valid: Kelas, Sumber Daya, Layanan, Jenis</p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>
DeliveryErrors	<p>Jumlah log acara penyebab CloudWatch Logs menerima kesalahan saat meneruskan data ke tujuan langganan. Jika layanan tujuan mengembalikan kesalahan yang dapat dicoba lagi, seperti pengecualian throttling atau pengecualian layanan yang dapat dicoba lagi (misalnya HTTP 5xx), CloudWatch Logs akan terus mencoba melakukan pengiriman hingga 24 jam. CloudWatch Logs tidak mencoba untuk mengirim ulang jika kesalahan adalah kesalahan yang tidak dapat dicoba lagi, seperti <code>AccessDeniedException</code> atau <code>ResourceNotFoundException</code>.</p> <p>Dimensi yang Valid: LogGroupName, DestinationType, FilterName</p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>

Metrik	Deskripsi
DeliveryThrottling	<p>Jumlah log acara penyebab CloudWatch Logs throttled saat meneruskan data ke tujuan langganan.</p> <p>Jika layanan tujuan mengembalikan kesalahan yang dapat dicoba lagi, seperti pengecualian throttling atau pengecualian layanan yang dapat dicoba lagi (misalnya HTTP 5xx), CloudWatch Logs akan terus mencoba melakukan pengiriman hingga 24 jam. CloudWatch Logs tidak mencoba untuk mengirim ulang jika kesalahan adalah kesalahan yang tidak dapat dicoba lagi, seperti <code>AccessDeniedException</code> atau <code>ResourceNotFoundException</code>.</p> <p>Dimensi yang Valid: <code>LogGroupName</code>, <code>DestinationType</code>, <code>FilterName</code></p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>
EMFParsingErrors	<p>Jumlah kesalahan penguraian yang ditemui saat memproses log format metrik tersemat. Kesalahan tersebut terjadi ketika log diidentifikasi sebagai format metrik tertanam tetapi tidak mengikuti format yang benar. Untuk informasi selengkapnya tentang format metrik tersemat, lihat Spesifikasi: Format metrik tersemat.</p> <p>Dimensi yang Benar: <code>LogGroupName</code></p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>

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 <code>MetricDatum</code> spesifikasi metrik yang disematkan. Untuk informasi tentang format metrik CloudWatch tertanam, lihat Spesifikasi: Format metrik tertanam . Untuk informasi tentang tipe <code>MetricDatum</code> , lihat MetricDatum di Amazon CloudWatch API Reference.
ErrorCount	<p>Note</p> <p>Kesalahan validasi tertentu dapat menyebabkan beberapa metrik dalam log EMF tidak dipublikasikan. Misalnya, semua metrik yang ditetapkan dengan ruang nama yang tidak valid akan dibatalkan.</p> <p>Dimensi yang Benar: <code>LogGroupName</code></p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p> <p>Jumlah operasi API yang dilakukan di akun yang mengakibatkan kesalahan.</p> <p><code>ErrorCount</code> adalah metrik penggunaan layanan CloudWatch Log. Untuk informasi selengkapnya, lihat CloudWatch Log metrik penggunaan layanan.</p> <p>Dimensi yang Valid: Kelas, Sumber Daya, Layanan, Jenis</p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>

Metrik	Deskripsi
ForwardedBytes	<p>Volume log acara dalam byte terkompresi yang diteruskan ke tujuan langganan.</p> <p>Dimensi yang Valid: LogGroupName, DestinationType, FilterName</p> <p>Statistik Valid: Sum</p> <p>Unit: Byte</p>
ForwardedLogEvents	<p>Jumlah log acara yang diteruskan ke tujuan langganan.</p> <p>Dimensi yang Valid: LogGroupName, DestinationType, FilterName</p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>
IncomingBytes	<p>Volume log acara dalam byte tak terkompresi yang diunggah ke CloudWatch Logs. Ketika digunakan dengan dimensi LogGroupName , ini adalah volume log acara dalam byte tak terkompresi yang diunggah ke grup log.</p> <p>Dimensi yang Valid: LogGroupName</p> <p>Statistik Valid: Sum</p> <p>Unit: Bita</p>
IncomingLogEvents	<p>Jumlah log acara yang diunggah ke CloudWatch Logs. Ketika digunakan dengan dimensi LogGroupName , ini adalah jumlah log acara yang diunggah ke grup log.</p> <p>Dimensi yang Valid: LogGroupName</p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>

Metrik	Deskripsi
LogEvents WithFindings	<p>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.</p> <p>Dimensi valid: Tidak Ada</p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>
ThrottleCount	<p>Jumlah operasi API yang dilakukan di akun Anda yang dibatasi karena kuota penggunaan.</p> <p>ThrottleCount adalah metrik penggunaan layanan CloudWatch Log. Untuk informasi selengkapnya, lihat CloudWatch Log metrik penggunaan layanan.</p> <p>Dimensi yang Valid: Kelas, Sumber Daya, Layanan, Jenis</p> <p>Statistik Valid: Sum</p> <p>Unit: Tidak ada</p>

Dimensi untuk metrik CloudWatch Logs

Dimensi yang dapat Anda gunakan dengan metrik CloudWatch Logs tercantum di bawah ini.

Dimensi	Deskripsi
LogGroupName	Nama grup CloudWatch log Logs untuk menampilkan metrik.
DestinationType	Tujuan langganan untuk data CloudWatch Logs, yang dapat berupa AWS Lambda, Amazon Kinesis Data Streams, atau Amazon Kinesis 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 setiap karakter yang tidak didukung akan diganti dengan tanda tanya (?).

CloudWatch Log metrik penggunaan layanan

CloudWatch Log mengirimkan metrik untuk CloudWatch melacak operasi API CloudWatch Log penggunaan. Metrik ini sesuai dengan kuota layanan AWS. Menelusuri metrik ini dapat membantu Anda mengelola kuota secara proaktif. Untuk informasi selengkapnya, lihat [Integrasi Service Quotas dan Metrik Penggunaan](#).

Misalnya, Anda dapat melacak ThrottleCount metrik atau menyetel alarm pada metrik tersebut. Jika nilai metrik ini naik, Anda harus mempertimbangkan untuk meminta peningkatan kuota untuk operasi API yang semakin dibatasi. Untuk informasi selengkapnya tentang kuota layanan CloudWatch Logs, lihat [CloudWatch Kuota log](#).

CloudWatch Log menerbitkan metrik penggunaan kuota layanan setiap menit di AWS/Logs ruang namaAWS/Usage dan ruang nama.

Tabel berikut mencantumkan metrik penggunaan layanan yang dipublikasikan oleh CloudWatch Logs. Metrik ini tidak memiliki unit tertentu. Statistik yang paling berguna untuk metrik ini adalah SUM, yang menunjukkan total operasi untuk periode 1 menit.

Masing-masing metrik ini diterbitkan dengan nilai untuk semuaService, ClassType, danResource dimensi. Mereka juga diterbitkan dengan dimensi tunggal 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 untuk menemukan metrik untuk API tertentu.

Metrik

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

Metrik	Deskripsi
	CallCount diterbitkan di keduaAWS/Usage danAWS/Logs ruang nama.
ErrorCount	Jumlah operasi API yang dilakukan di akun yang mengakibatkan kesalahan. ErrorCount diterbitkan hanya dalamAWS/Logs.
ThrottleCount	Jumlah operasi API yang dilakukan di akun Anda yang dibatasi karena kuota penggunaan. ThrottleCount diterbitkan hanya dalamAWS/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 nilaiResource.
Service	Nama dari layanan AWS yang berisi sumber daya. Untuk metrik penggunaan CloudWatch Logs, nilai untuk dimensi ini adalahLogs.
Class	Kelas sumber daya yang ditelusuri. CloudWatch Metrik penggunaan API Logs menggunakan dimensi ini dengan nilaiNone.
Type	Jenis sumber daya yang ditelusuri. Saat ini, ketika Service dimensi 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 standar
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	5 transaksi per detik (TPS/akun/Wilayah), setelahnya transaksi tersebut akan mengalami throttling. Anda dapat meminta kenaikan kuota.
CreateLogStream	50 transaksi per detik (TPS/akun/Wilayah), setelahnya transaksi tersebut akan mengalami throttling. Anda dapat meminta penambahan kuota.
DeleteLogGroup	5 transaksi per detik (TPS/akun/Wilayah), setelahnya transaksi tersebut akan mengalami throttling. Anda dapat meminta kenaikan kuota.
DeleteLogStream	5 transaksi per detik (TPS/akun/Wilayah), setelahnya transaksi tersebut akan mengalami throttling. Anda dapat meminta kenaikan kuota.
DescribeLogGroups	5 transaksi per detik (TPS/akun/Wilayah). Anda dapat meminta penambahan kuota.
DescribeLogStreams	5 transaksi per detik (TPS/akun/Wilayah). Anda dapat meminta kenaikan kuota.

Sumber Daya	Kuota standar
Bidang log yang ditemukan	<p>CloudWatch Wawasan Log dapat menemukan maksimal 1000 bidang peristiwa log dalam grup log. Kuota ini tidak dapat diubah.</p> <p>Untuk informasi selengkapnya, lihat Log yang didukung dan bidang yang ditemukan.</p>
Bidang log yang diekstraksi dalam log JSON	<p>CloudWatch Wawasan Log dapat mengekstrak maksimal 200 bidang peristiwa log dari log JSON. Kuota ini tidak dapat diubah.</p> <p>Untuk informasi selengkapnya, lihat Log yang didukung dan bidang yang ditemukan.</p>
Tugas ekspor	Satu tugas ekspor aktif (berjalan atau tertunda) pada satu waktu, per akun. Kuota ini tidak dapat diubah.

Sumber Daya	Kuota standar
FilterLogEvents	<p>25 permintaan per detik di AS Timur (Virginia Utara).</p> <p>10 permintaan per detik di Wilayah berikut:</p> <ul style="list-style-type: none">• AS Timur (Ohio)• AS Barat (California Utara)• AS Barat (Oregon)• Afrika (Cape Town)• Asia Pasifik (Hong Kong)• Asia Pasifik (Mumbai)• Asia Pasifik (Seoul)• Asia Pasifik (Singapura)• Asia Pasifik (Tokyo)• Asia Pasifik (Sydney)• Kanada (Pusat)• Eropa (Irlandia)• Eropa (London)• Eropa (Milan)• Eropa (Paris)• Eropa (Stockholm)• Timur Tengah (Bahrain)• Amerika Selatan (Sao Paulo)• AWS GovCloud (AS-Timur)• AWS GovCloud (AS-Barat) <p>5 permintaan per detik di semua Wilayah lainnya.</p> <p>Kuota ini tidak dapat diubah.</p>

Sumber Daya	Kuota standar
GetLogEvents	<p>30 permintaan per detik di Eropa (Paris).</p> <p>25 permintaan per detik di Wilayah berikut:</p> <ul style="list-style-type: none">• AS Timur (N. Virginia)• AS Timur (Ohio)• AS Barat (California Utara)• Afrika (Cape Town)• Asia Pasifik (Hong Kong)• Asia Pasifik (Mumbai)• Asia Pasifik (Seoul)• Asia Pasifik (Singapura)• Asia Pasifik (Tokyo)• Asia Pasifik (Sydney)• Kanada (Pusat)• Eropa (London)• Eropa (Milan)• Europe (Stockholm)• Timur Tengah (Bahrain)• Amerika Selatan (Sao Paulo)• AWS GovCloud (AS-Timur)• AWS GovCloud (AS-Barat) <p>10 permintaan per detik di semua Wilayah lainnya.</p> <p>Kuota ini tidak dapat diubah.</p> <p>Kami merekomendasikan langganan jika Anda terus memproses data baru. Jika Anda membutuhkan data historis, sebaiknya Anda mengekspor data ke Amazon S3.</p>

Sumber Daya	Kuota standar
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.
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.
<u>PutLogEvents</u>	Ukuran batch maksimum PutLogEvents permintaan adalah 1MB. 800 transaksi per detik per akun per Wilayah, kecuali untuk Wilayah berikut dengan kuota 1500 transaksi per detik per akun per Wilayah: US East (N. Virginia), US West (Oregon), dan Europe (Irlandia). Anda dapat meminta peningkatan kuota throttling per detik dengan menggunakan layanan ini. Service Quotas
Batas waktu eksekusi kueri	Waktu kueri di CloudWatch Logs Insights habis setelah 60 menit. Batas waktu ini tidak dapat diubah.

Sumber Daya	Kuota standar
Grup log yang dikueri	Maksimal 50 grup log dapat ditanyakan dalam satu kueri Wawasan CloudWatch Log. Kuota ini tidak dapat diubah.
Konkurensi kueri	Maksimal 30 kueri Wawasan CloudWatch Log bersamaan, termasuk kueri yang telah ditambahkan ke dasbor. Kuota ini tidak dapat diubah.
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 PutQueryDefinition tidak 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.

Sumber Daya	Kuota standar
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 LayananAWS .
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
<u>CloudWatch Log menambahkan dukungan sintaks pola filter ekspresi reguler untuk Live Tail</u>	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 <u>Memfilter sintaks pola</u> di Panduan Pengguna CloudWatch Log Amazon.	13 November 2023
<u>CloudWatch Log menambahkan dukungan sintaks pola filter ekspresi reguler untuk filter metrik, filter langganan, dan peristiwa log filter</u>	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 <u>Memfilter sintaks pola</u> di Panduan Pengguna CloudWatch Log Amazon.	5 September 2023
<u>CloudWatch Log Insights menambahkan perintah pola</u>	Sekarang Anda dapat menggunakan pola dalam kueri Wawasan CloudWatch Log untuk secara otomatis mengelompokkan data log	Juli 17, 2023

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.

[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

CloudWatchLogsRead OnlyAccess kebijakan 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

<u>CloudWatch Log Insights dirilis</u>	Anda dapat menggunakan Wawasan CloudWatch Log untuk mencari dan menganalisis data log secara interaktif. Untuk informasi selengkapnya, lihat <u>Menganalisis Data CloudWatch Log dengan Wawasan Log</u> di Panduan Pengguna CloudWatch Log Amazon.	27 November 2018
<u>Dukungan untuk titik akhir VPC Amazon VPC</u>	Anda sekarang dapat membuat koneksi pribadi antara VPC dan CloudWatch Log Anda. Untuk informasi selengkapnya, lihat <u>Menggunakan CloudWatch Log dengan Titik Akhir VPC Antarmuka</u> 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 <u>Menggunakan CloudWatch Log dengan titik akhir VPC antarmuka</u> .	Selasa, 07 Maret 2018
Log kueri DNS Route 53	Anda dapat menggunakan CloudWatch Log untuk menyimpan log tentang kueri DNS yang diterima	7 September 2017

Perubahan	Deskripsi	Tanggal rilis
	oleh Route 53. Untuk informasi selengkapnya, lihat Apa itu Amazon CloudWatch Logs? atau Mencatat Log Kueri DNS dalam Panduan Developer Amazon Route 53.	
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
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 Membuat CloudWatch log panggilan API Amazon LogsAWS 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

Perubahan	Deskripsi	Tanggal rilis
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
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.